SIMULATION OF PHYSICAL SYSTEMS WITH FEW DEGREES OF FREEDOM

by

Engin Kandıran

Submitted to Graduate School of Natural and Applied Sciences

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Physics

Yeditepe University

2019

SIMULATION OF PHYSICAL SYSTEMS WITH FEW DEGREES OF FREEDOM

APPROVED BY:

Prof. Dr. Avadis Simon Hacınlıyan
(Thesis Supervisor)
(Yeditepe University)

Prof. Dr. Şerife İpek Karaaslan
(Yeditepe University)

Assist. Prof. Dr. Hacı Ahmet Yıldırım
(Sakarya University)

Assist. Prof. Dr. Gökhan Şahin
(Yeditepe University)

Assist. Prof. Dr. Nazım Ziya Perdahçı
(Mimar Sinan Fine Arts University)

DATE OF APPROVAL: .... /.... /2019

ACKNOWLEDGEMENTS

During this dissertation process, I have gained great experience and a great deal of support and assistance. Particularly, I would first like to thank my thesis supervisor, Prof. Dr. A.Hacınlıyan, who give endless support and guidance in the whole research topics in my thesis during the whole doctorate phases. I need to clearly state that I will not get this degree without his expert guidance and his academic trust in me.

I would also send a special thanks to Prof. Dr. İpek Karaaslan, who always suppored me in many technical and academic problems during PhD process.

Finally, I would like to express my eternal gratitude to my parents who are in with me during all difficult situation during doctoral process and who do not lose their faith in me in all situation.

ABSTRACT

SIMULATION OF PHYSICAL SYSTEMS WITH FEW DEGREES OF FREEDOM

In physics, the models that describe our universe from macro to micro scales involve a set of equations which denotes the relationships between variables. These systems are generally represented by differential equations and are classified as dynamical systems. Dynamical systems are studied in many science fields. With the use of chaos theory within dynamical systems, many hitherto unknown properties of the systems have been discovered. Most real systems are formalized under Hamiltonian mechanism and Hamiltonian systems both display regular and chaotic behavior. In this study, we investigate a Hamiltonian system, namely Matinyan-Yang-Mills-Higgs system, and we try to investigate the regions where the system makes transition from regular to chaotic regions. In the second part of the thesis, we study the power of artificial neural networks in modeling dynamical systems and estimating nonlinear time series. In the third part, we use dynamical system as random number generators and we use these random number generators in image encryption.

ÖZET

AZ SAYIDA SERBESTLİK DERECESİNE SAHİP FİZİKSEL SİSTEMLERİN SİMÜLASYONU

Fizikte, evrenimizi, makrodan mikro düzeye kadar tanımlayan modeller, değişkenler arasındaki ilişkileri ifade eden denklemler listesi şeklinde düzenlenmiştir. Bu sistemler genellikle diferansiyel denklemlerle temsil edilir ve dinamik sistemler olarak sınıflandırılır. Dinamik sistemler birçok bilim alanı tarafından incelenmektedir. Kaos teorisinin dinamik sistem içerisinde kullanılmasıyla, sistemlerin daha önce pek bilinmeyen özellikleri keşfedilmiştir. Gerçek sistemlerin çoğu Hamiltonyen mekanizması altında biçimlendirilir ve Hamiltonyen sistemler hem düzenli hem de kaotik davranış gösterir. Bu çalışmada bir Hamiltonyen sistem, Matinyan-Yang-Mills-Higgs sistemini inceledik ve sistemin düzenli bölgeden kaotik bölgelere geçiş yaptığı koşulları araştırdık. Tezin ikinci bölümünde, yapay sinir ağlarının dinamik sistemleri modellemede ki ve lineer olmayan zaman serilerindeki tahmin etme gücünü araştırdık. Üçüncü bölümde ise, dinamik sistemleri rassal sayı üreteci olarak kullandık ve bu rassal sayı üreteclerini görüntü şifrelemede kullandık.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

b_i	bias of neuron j
m	embedding dimension
J	Jacobian
$\hat{o_j}$	output of neuron j
rmse	root-mean-square error
W_{ij}	weight of the connection from neuron i to j
λ	eigenvalue
η	noise level
$\mathbb{R}$	Real Numbers
CPRNG	Chaotic Random Number Generator
FFN	Fuzzy Neural Networks
FN	Feedforward Network
H	Hamiltonian
K	transformed perturbed Hamiltonian
LCG	Linear Congruential Generator
MET	Multiplicative Ergodic Theorem
MFN	Multilayer Feedforward Network
MYMH	Matinyan-Yang-Mills-Higgs
NARX	Nonlinear auto regressive model with external input.
PB	Poisson Bracket
PRNG	Pseudo Random Number Generator
RBF	Radial Basis Function
RNG	Random Number Generator
TRG	True Random Number Generator

1. INTRODUCTION

Computers have become the integral part of todays scientific studies and have had a great impact on the development of the recent knowledge in the scientific literature from mathematics to biology. In physics, through the advances in the computational power of computers, researchers have been able to make great contributions to our understanding of nature with high accuracy and speed. In physics, the main aim is to develop models of nature to make the universe more understandable and computers allow physicists to make simulations of constructed models of nature visible and testable. In addition to this, the models can be tested in extreme cases to make predictions about the future of systems which are under study.

Computer simulation have become the integral scientific tool especially in meteorology and nuclear physics in the period following the World War II, and it has been used as an indispensable tool in other disciplines of physics. Some of the scientific disciplines which use computer simulation extensively are: astrophysics, particle physics, materials science, engineering, fluid mechanics, climate science, evolutionary biology, ecology, economics, decision theory, medicine, sociology, epidemiology, and many others. There are also some other disciplines, such as chaos theory and complexity theory, whose development process have gained impulse alongside the development of the computational models they study.

While talking about computer simulations, a more clear definition of computer simulation may be required. However, there is no-single accepted definition of it which can be done easily. Yet, for physicists′ point of view, computer simulation can be defined as: a program that is run on a computer and that uses step-by-step methods or stochastic methods to explore and understand the approximate behavior of real-world systems (the system will also be hypothetical ones). On the other hand, computer simulations can be categorized in two classes in terms of their usage purpose: for predicting the data which is not available and for understanding the system according to data that we have. Computer simulations are used to estimate how the real-world system would behave under a given set of particular conditions. This is called as prediction and predictions can be made for detecting future cases

of examined system as well as past phases of it. By use of simulations we can make different types of predictions such as point prediction, interval estimation and global or systematic predictions. One example of point predictions (or estimates) is prediction of the next-day stock market indices value. Suppose we want to look for the range of parameter values in which a dynamical system, such as the Lorenz system, displays chaotic behavior, we need interval prediction via computer simulations. Systematic reductions are useful to answer questions like: Is the orbit of investigated planet stable or not? Or what is the Lyapunov exponent or fractal dimension of the studied time series?

Up to now, application areas and benefits of computer simulations, especially in physics, were discussed. As in the case of other approaches, computer simulations have their own disadvantages or drawbacks. As we already mentioned models that are generated to simulate real-world phenomena will be very helpful in understanding of the phenomena under consideration but simulated models can give misinterpretation of data or specious (inadequate) information or conclusions about the system since physical models that we use may not be sufficient to describe the full dynamics of the real system. For instance, it is not easily possible to predict the occurrence and effects of earthquakes and tsunami. Another drawback of computer simulations is: Simulations, especially complex ones, may need the use of a computer system with a high processing power and enormous memory usage. The most important factor which can be seen as a negative point for simulation in physics is round-off errors and uncertainties in computer simulations.

Chaos theory is one of the interdisciplinary study fields that uses computer simulations in many applications in its scope. Although, the term chaos in general is understood as *state of disorder*, in theoretical point of view, it is related to determinism and extent of predictability, at least in principle, of future outcomes of the system of interest. Many definitions of chaos theory can be given but in general sense, chaos theory is interested in finding reasonable explanation, in deterministic point of view, to complex behavior of *dynamical system(s)*. Dynamical systems are sets of equations (mostly nonlinear ones) which describe the evolution of its parameters with respect to time or time parameter. Variables of the dynamical system create a set of vectors in finite or infinite dimensional space and these vectors form the *phase space*. Dynamical systems are the simple mathematical models which

try to explain dynamics of real-world system, such as population dynamics (predator-prey systems), or they can be heuristic models whose equations can behave unpredictably. Chaos theory especially deals with dynamical systems whose dynamics sensitively depend on initial conditions which means according to small changes in initial conditions (or perturbations), systems can display widely varying behavior.

To classify the dynamical system as chaotic, it should satisfy some certain criteria. Although chaotic systems display random like behavior, they display some common underlying patterns such as self-similarity, fractal like behavior and sensitive dependence on initial conditions (as it is already stated.) What makes these systems irregular is the fact that the behavior is different from that of linear systems we are accustomed to. However, to classify a dynamical system as chaotic, there should be some criteria. In [1], Delaney and Eckmann stated the following criteria which are necessary to define dynamical systems as chaotic:

- It should be sensitive to initial conditions.
- It must display topological mixing.
- It should have dense periodic orbits

We have already mentioned about sensitive dependence on initial conditions but in society it is generally known as *butterfly effect* which is originated from the Lorenz model (or attractor). In 1963, Edward Lorenz published his famous Lorenz model which describes atmospheric convection. Lorenz system is a nonlinear mathematical model which describes the relations with change of three quantities with respect to time: the rate of convection, the horizontal flow speed and temperature variation, and the vertical temperature variation. Making changes on the parameters of Lorenz system or slightly changing initial conditions on the system results in different outcomes and these aspects of analysis of Lorenz System has been studied by many researchers [2, 3]. Because of the sensitivity to initial conditions there is limited amount of information about the system (as is usually the case in practice), then after a certain time the system does not have long predictable time horizon. The most typical real-life cases of this short-term predictability is the weather. Because of this reason, weather predictions have been limited by a certain timeline, usually one week.

Most of the time, topological mixing (transitivity) can be ignored during chaos classification and in general sensitivity to initial condition is accepted as the only parameter which displays chaotic behavior. However, there are many cases that are sensitive to initial conditions but do not give chaos. Physical meaning of topological mixing is the following: overlapping open set of phase space of investigated system with any other region will lose its dependence on region when the system evolves for sufficiently long time. The third condition to determine the chaotic behavior of a dynamical system is having dense periodic orbit and this means that every point in the phase space is approached arbitrarily closely by periodic orbits if sufficient time elapses.

Chaotic dynamical systems have been a very popular topic in recent years and its results are used especially by physicists to understand the dynamics of real-world phenomena especially where finding the exact solutions is impossible or very hard. For instance, to understand the early stages of the universe, nonlinearities in the general theory of relativity [4] leads to chaos and solutions to attack this problem are done by numerical approaches. Dynamics of the real physical events are described by Hamiltonian formalism in both classical and quantum physics. However, most of the Hamiltonian systems are non-integrable. Non-integrability in Hamiltonian systems generally originated from nonlinear terms in the system. The first example of non-integrable chaotic system is discovered by Poincaré [5] via geometric analysis in phase space. Koopman and von Neumann discovered the first analytical evidences of chaos in Hamiltonian system with two-degrees of freedom [6, 7, 8]. The most commonly observed structure in integrable Hamiltonian systems is invariant tori. In this geometric view, the trajectories that start on any torus stays on the same torus for all time. If we have enough information about the torus, this information can be used to analyze the stability of the dynamical system or other chaotic behavior of the system. If the Hamiltonian system is nearly integrable, the chaotic and regular motions can be both observed. As the Hamiltonian becomes more non-integrable, the chaotic areas grow in the phase space and start to merge with each other, so the invariant tori become destroyed.

Up to know, a brief introduction about the chaos theory, its applications areas and role of computer simulation in studying dynamical system is presented. In addition to this, the three criteria to define a dynamical system as chaotic are described. In this sense, it is

necessary to mention about the methods to find out the chaos in dynamical systems. The most common measure of the chaos is the Lyapunov exponent(s). Lyapunov exponents measure the rates of convergence or divergence of nearby trajectories from slightly perturbed initial conditions [9, 10], and this concept is found by, as its name suggest, Lyapunov [11]. If the dynamical system is N-dimensional, we have N-Lyapunov exponents. At this point, it is necessary to mention about the significance of Lyapunov exponent(s). If there exist a positive Lyapunov exponent, signature of chaos is observed. If there exist a zero Lyapunov exponent, the system can be said in steady state along that direction and the system can be classified as conservative. In the case of negative Lyapunov exponent, we can consider the system dissipative or non-conservative. However, suppose that we are dealing with N-dimensional system, that's why we have N-exponents, and some exponents are negative and some of them are positive. In that case, if more than one exponent is positive then we have a hyper-chaotic system. Yet, it is important to note here that if the system is Hamiltonian one, sum of all exponents must be equal to zero according to Liouville's theorem which is going to be mentioned later.

In the literature, practical ways of calculating Lyapunov exponents are widely discussed. The theoretical study on existence of Lyapunov exponents started with Osedelec[12] and in this paper, Multiplicative Ergodic Theorem (MET) is given which allows the theoretical basis for the numerical computation of Lyapunov exponents. Ergodic theory has also been studied by many researchers such as Ruelle[13]. There are many numerical algorithms based on theoretical approach to find the Lyapunov exponents in literature. The most commonly used method is the Wolf-Benettin algorithm [10]. There are two versions of Wolf-Benettin algorithm: one of them for time series and in the other version one can find the Lyapunov exponents through differential equations of the system augmented by variational equations. Another important algorithm is the Wiesel algorithm [14, 15] where evolution equations are written for the unit vectors along the fiducial trajectory. Rosenstein and Kantz developed their own algorithms for calculating largest Lyapunov exponents in [16] and [17] respectively for time series.

Another important method which is used for determining chaos in a dynamical system is stability analysis. By using the stability analysis, the fixed point of a dynamical system

can be classified as stable (attractor), structurally stable(focus) or unstable (repeller). If the system is perturbed by a small amount around the unstable fixed point (or repeller), the trajectory of the system grows in time. If the fixed point is stable, the trajectory of the system shrinks toward the fixed point. Stability analysis is easy for one-dimensional system but for N-dimensional systems, where N$\geq$2, the stability analysis is more complicated. For these cases (N$\geq$2), the fixed points are classified by studying at eigenvalues of the Jacobian matrix. By looking at the signs of eigenvalues, the fixed points can be classified as stable or unstable. In addition to this, if the eigenvalues are complex, the fixed points are can be classified as spiral or limit cycles. Detailed discussion of stability analysis is going to be given in the chapter 2. Since the Jacobian matrix also evolves with the system, this is used as a basis for the calculation of Lyapunov exponents in the Ruelle-Eckmann algorithm [13].

Up to now, numerical methods to calculate Lyapunov exponents are discussed. In this study, one of the main concern is related to using artificial neural networks in computation of chaotic parameters and capabilities of neural networks in estimations of chaotic time series. Previously mentioned numerical methods fail in terms of accuracy in calculating Lyapunov exponents due to the following reasons: the estimated exponents are sensitive to number of data points and noise in the data set. In addition to estimating chaotic parameters, modeling a chaotic system or time series is an important topic in literature. All forecasting methods to predict evolution of a system are based on past measurement of time series in hand. In the case of nonlinear system with many degrees of freedom, there is no simple procedure to find a simple set of solutions without some sort of assumptions. This kind of simplifications allow us to reduce the number of degree of freedom of the system up to relevant degree. In that case, the dynamics of the system can be considered in a subspace Γ_S, of space Γ, which is called as *attractor* with lower dimension. In this kind of situation, neural networks can be considered a very successful candidate tool for forecasting. In literature, there are many studies related to finding Lyapunov exponents with neural networks. In [18], R.Gencay and W.D.Dechert present an algorithm which uses feedforward neural network to calculate the Lyapunov exponents of n-dimensional dynamical system which is based on Ruelle-Eckman algorithm [13]. In another study [19], capabilities of neural networks in estimating Lyapunov spectrum are compared with the exponent values calculated by TISEAN package [17]. The filtering capacities of neural network with noisy chaotic time series is discussed in [20] and

it is shown that neural networks are very efficient tool in prediction of noisy time series. In addition to feedforward neural networks other types of networks are used in literature for chaotic time series and dynamical systems such as recurrent neural networks, radial basis function (RBF) neural network, fuzzy neural networks (FNNs) etc... In [21], fuzzy neural network is used for application to chaotic time series prediction and it is reported that fuzzy neural networks bring advantages by reducing the training time and reducing the dimension of the system. Recurrent neural network is used in nonlinear modeling [22] and forecasting performance of it is compared with feedforward neural networks. In addition to estimating Lyapunov exponents and forecasting chaotic time series, neural networks are used to determine other chaos parameters such as embedding dimension [23].

In field of cyber-security, chaos theory has found its application. Due to unpredictable nature of chaotic systems, in image encryption, secure communication many scientist use chaotic system in security algorithms. Also generating random numbers is very important topic especially in Monte-Carlo simulations where absolute randomness is required. Ideal random number generator should have long frequency and it should be fast. These properties can be satisfied by chaotic systems.

In this dissertation, we try to find answers to 3 main problems: as a starting point we work on a Hamiltonian physical model, namely the Matinyan-Yang-Mills-Higgs system, which may offer answers to the dynamics of our early universe and try to investigate the possible regions of chaos and transition regimes from stability to instability or vice versa. Secondly, we make use of neural networks in modeling dynamical systems, estimating chaotic parameters such as Lyapunov exponents and estimating nonlinear time series from previous states of them. In chapter 5, we try to construct random number generator by using two different well-known dynamical system and test their usability in image encryption.

2. THEORETICAL BACKGROUND RELATED TO CHAOS THEORY AND ANALYSIS OF MATINYAN-YANG-MILLS-HIGGS SYSTEM

2.1. DEFINITIONS ABOUT DYNAMICAL SYSTEM THEORY

Although the nature of the physical laws of microscopic scales are not completely deterministic, due to their probabilistic nature which is explained by quantum mechanics, in the macroscopic word, classical solution, that are used to understand the realism of event in our universe, yields very accurate descriptions such as in celestial bodies.

2.1.1. Lyapunov Exponent

In chapter 1, the basic definition of Lyapunov exponents is already given. In this section, the mathematical definition of Lyapunov exponent(s) is going to be introduced and geometric meaning of it is discussed.

Lyapunov exponents give the measure of the rate of divergence or convergence of nearby trajectories. In Fig. 2.1, geometric point of view is presented. Start with considering the given nearby points X_0 and $X_0 + \Delta X_0$ in Fig. 2.1 which are going to create the orbit or trajectories of the investigated dynamical system based on equations of the system.

The generated orbits are functions of parameters and variables of the system and time. If one of these orbits is selected as a reference orbit (fiducial trajectory), the separation between the two orbits can be defined as a function of time. It is also necessary to note that this separation is also a function of initial point $\Delta X(X_0, t)$. Since there is a time dependence in the calculation of separation between these two nearby trajectory, it will give local rate of convergence (or divergence). Quantitatively, two trajectories in phase space with initial separation converge or diverge with rate $|\Delta X(t)| \approx e^{\lambda t} \Delta X_0$ where λ is Lyapunov

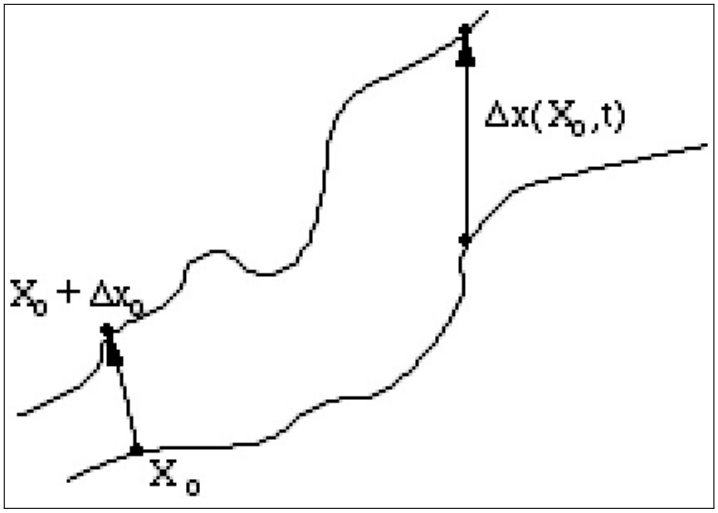


Figure 2.1. Geometric Interpretation of Calculation of Lyapunov Exponent.

exponent.As t goes to infinity, we can have the global Lyapunov exponent λ given by Eq. 2.1:

$$\lambda = \lim_{t\to\infty} \lim_{\Delta X_0 \to 0} \frac{1}{t} In \left| \frac{\Delta X(X_0, t)}{X_0} \right| \tag{2.1}$$

It is necessary to note that if the system is N-dimensional, then there exist N-Lyapunov exponents. In other words, Lyapunov exponent is exponential rate of convergence of trajectories, but in our consideration its physical interpretation is more important. If the Lyapunov exponent is $\lambda < 0$, then nearby trajectories converge (or attract) to stable fixed point or stable periodic orbit. One of the examples from physics is critically damped oscillation where the system is attracted towards the equilibrium point.

The second case is $\lambda = 0$. In that case, since the rate of convergence is zero, the system can be considered as in steady state mode. Another important interpretation of zero Lyapunov exponent is that the system has a conserved quantity and if the system is Hamiltonian, then the system is conservative. In this point, we need to mention about Lyapunov stability issue which is done in later sections.

The last possibility is $\lambda > 0$. This is the most critical case for chaos in dynamical system since it shows the chaotic behavior in the system. However, it is meaningful to ask whether positive Lyapunov exponents always refers to chaos or not. As it can be understood from

previous discussions, the Lyapunov exponents quantify the sensitivity to initial conditions. While calculating Lyapunov exponents, the recorded or generated time series are used and this time series include some level of noise. Because of the noise effect, the system has a positive Lyapunov exponent which will not display the correct behavior of the system.

For N-dimensional dynamical system, the *Lyapunov exponent spectrum* should be mentioned. For a N-dimensional dynamical system with evolution equations $\dot{x_i} = f_i(x)$, the Lyapunov spectrum is defined as $\{\lambda_1, \ldots, \lambda_N\}$, which depends on initial point X_0. Behavior of vectors in the tangent space of phase space described by the Jacobian matrix J whose elements are defined as:

$$J_{ij}(t) = \left.\frac{df_i(x)}{dx_j}\right|_{x_t} \tag{2.2}$$

The Jacobian matrix shows the effect of small changes on the initial point X_0 on the final point $f_i(x_0)$. The following limit:

$$\lim_{t\to\infty} (J.J^T)^{\frac{1}{2t}} \tag{2.3}$$

where J^T transpose of Jacobian matrix, defines a matrix $\Lambda(X_0)$ whose existence is guaranteed by Osedelec's Theorem [12]. Then the Lyapunov exponents of the system are the eigenvalues of $\Lambda(X_0)$ matrix. Osedelec's multiplicative ergodic theorem gives theoretical background of calculation of Lyapunov exponents.

2.1.1.1. Largest Lyapunov Exponent

The usual indicator of chaos is the largest Lyapunov exponent. If the equations of the system which generates the chaotic trajectory are known, calculating the largest Lyapunov exponent follows a simple procedure. A basic idea behind this calculation is the following: consider the two nearby trajectory and calculate their average logarithmic rate of separation.

When these two nearby orbits separate from each other, one of them must turn back to the

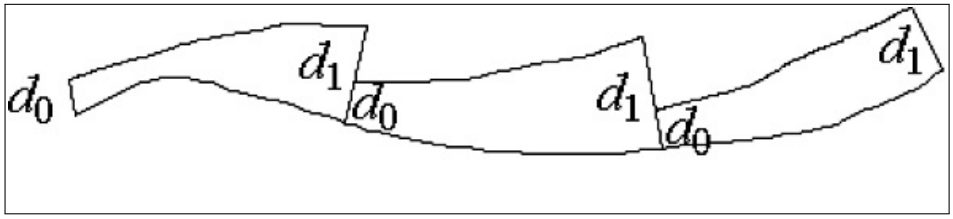


Figure 2.2. Geometric View of Calculation of Largest Lyapunov Exponent.

vicinity of the other along the line of separation. This conservative procedure is composed of the following steps:

- Choose a starting point at the basin of attraction.
- Make iteration until the orbit is on the attractor.
- Choose a nearby point separated by distance d_0 as it is shown in Fig. 2.2.
- Iterate both orbits one time and recalculate the new separation distance d_1.
- Calculate $ln|d_0/d_1|$
- Rearrange one of the orbits so that their separation is d_0 and is in the same direction at d_1.
- Follow all the previous steps for many times.
- Take the average of each iteration.

Although the calculation of largest Lyapunov exponent seems to be easy, the calculation of it is prone to some error and mistakes. Choosing a starting point at the basin of attraction requires a previous knowledge about the system so that if a good initial point is not chosen, then the calculated exponent will mislead us. Another possible error in this method happens when d_1 is very close to zero. When we take a natural logarithm, our calculation includes the error. Furthermore, as many steps are taken, error on the calculation of exponent may accumulate.

Kantz[17] and Rosenstein[16] develop their own algorithms to calculate largest Lyapunov exponent numerically.

2.1.2. Lyapunov Stability and Lyapunov Function

In physics, for most of problems finding equilibrium points of the system is very important where the stability plays role. For dynamical system theory, finding equilibria of the system on hand give clues about dynamics of system.

Consider the following autonomous(time-independent) system:

$$\dot{x} = f(x(t)) \tag{2.4}$$

If x^* is fixed point of the system, it satisfies $f(x^*) = 0$. Then, we can write Eq. 2.4 in series expansion form close to x^* as:

$$\dot{x} = f(x^*) + \left.\frac{\partial f}{\partial x}\right|_{x=x^*} (x - x^*) + \cdots = \left.\frac{\partial f}{\partial x}\right|_{x=x^*} (x - x^*) + \dots \tag{2.5}$$

Then the partial derivatives in right hand side of Eq. 2.5 creates a matrix, called as Jacobian J

$$J = \left.\frac{\partial f}{\partial x}\right|_{x^*} \tag{2.6}$$

Then the system in Eq.2.4 can be reduced and analyzed in linearized form:

$$\dot{\overline{x(t)}} = J\overline{x(t)} \tag{2.7}$$

with $\overline{x} = x - x^*$. The eigenvalues of J give information about the stability of the fixed point x^* as follows:

- If the eigenvalues of J evaluated at x^* have all negative real parts, the fixed point *asymptotically stable* for system 2.4.
- If at least one of the eigenvalues of J has positive real part, then the fixed point x^* is *unstable*.
- If the none of the eigenvalues of J has real part(purely imaginary), linearized system

does not give information to characterize the stability of point x^*.

- If J has complex conjugate eigenvalues, the equilibrium point x^* is called a stable or unstable spiral focus, depending on the sign of the real part.

This kind of analysis is generally called as the first method of Lyapunov stability. Lyapunov also states a second method to define stability. The second method also called as *Lyapunov stability criterion* as stated as follows: Suppose there exist a function $V(x)$, where x states variables of system defined in Eq. 2.4, which satisfies the following properties:

$$\begin{aligned} V(x^*) &= 0 \quad \text{iff} \quad x = x^* \\ V(x^*) &> 0 \quad \text{for} \quad x \neq x^* \end{aligned} \tag{2.8}$$

In addition to given conditions in Eq. 2.8 if $V(x)$ satisfies:

$$\frac{dV(x)}{dt} \leq 0, \quad \forall t \tag{2.9}$$

Then if function $V(x)$ is called as *Lyapunov function* and the system considered as Lyapunov stable. However, for most of the cases, it is not easy a construct a Lyapunov function $V(x(t))$.

2.2. POINCARÉ SECTION (MAP) ANALYSIS

Poincaré section analysis is a very useful approach in investigating the phase space structure and time evolution of a dynamical system, especially for nearly non integrable systems. This method has become very popular in the last decades in connection with the KAM theorem for weakly perturbed Hamiltonian systems. More specifically, this method is very efficient if one wants to explore the system dynamics with two degrees of freedom. Consider the following Hamiltonian system:

$$H(p_1, p_2, q_1, q_2) = E \tag{2.10}$$

This restricts the orbit to a three-dimensional hyper-surface (or energy surface) in phase space. Suppose that the system has a second integral of the motion I_2.

$$I_2(p_1, p_2; q_1, q_2) = C \tag{2.11}$$

where C is constant. Then, the orbits of the system are embedded in a two-dimensional surface which is the intersection of the three-dimensional energy surface E and second integral of motion C. For instance, we can express the generalized momentum p_1 with respect to generalized coordinates q_1 and q_2 as $p_1 = p_1(q_1, q_2, E, C)$. Then consider the surface where $q_2 = 0$, the trajectory of the motion embeds in a one dimensional curve. In general, suppose for a given Hamiltonian of the form in Eq.2.10, it is not known whether or not the second integral I_2 exists. We can observe its existence via solving the Hamilton's equations of motion numerically:

$$\begin{aligned} \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i} \\ \frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} \end{aligned} \tag{2.12}$$

for $i = 1, 2$. Then the plot of p_2 vs. q_2 for initial values of $p_1 \geq 0$ and $q_1 = 0$ is the Poincaré section. If the system is integrable, the series of points seem to lie in a curve. If the system is non-integrable, the points of trajectory seem to scatter around a finite region due to energy conservation.

2.3. METHOD OF AVERAGING

Averaging method is a useful classical method for analyzing weakly nonlinear problems. The origin of method of averaging goes back to 1788 in which the Lagrangian formulation of the gravitational three body problem was treated in terms of perturbation of the two body problem. However, up to 1930s, no systematic proof of the method of averaging was given. Krylov and Bogoliubov gave the systematic study on the method of averaging in [24, 25]. Then the method of averaging became one of the standard mathematical methods for approximate analysis of oscillatory processes in non-linear dynamics. The importance of the averaging theory is that the averaged system would be a good candidate

for approximation of the exact dynamics of the original system. This is very useful in cases which classical perturbation theory fails to explain.

The method of averaging can be used for dynamical systems in the following form:

$$\dot{x} = \epsilon f(x, t, \epsilon) \tag{2.13}$$

where $x \in U \subseteq \Re^n$, $\epsilon << 1$ and $f : \Re^n \times \Re \times \Re^+$ is C^r, $r \geq 1$ and of the period $\tau > 0$ in t, U is bounded and open. According to this, values in the averaged system are defined as:

$$\dot{y} = \epsilon \frac{1}{\tau} \int_0^\tau f(y, t, 0)dt = \epsilon \bar{f}(y) \tag{2.14}$$

We have said that the averaged system is a good approximation of the original system but it is necessary to show that qualitative properties of the solutions of the averaged system correspond to those of the original system. To justify this we need to recall the Averaging theorem given in[26]. A paraphrase of the relevant part of theorem is following:

Theorem 1 (Averaging Theorem)**.** *Eq. (2.13) can be cast the form Eq. (2.15) using the C^r near identity transformation $x = y + \epsilon w(y, t, \epsilon)$ under which Eq. (2.13) becomes*

$$\dot{y} = \epsilon f(y) + \epsilon^2 f_1(y, t, \epsilon) \tag{2.15}$$

where f_1 has period τ in t. Furthermore,

- *i If two solutions of Eq.(2.13) and Eq.(2.14), starting from initial conditions x_0,y_0, with $|x_0 - y_0| = O(\epsilon)$, then $|x_0 - y_0|$ remains $O(\epsilon)$ on a time scale $t \sim 1/\epsilon$.*
- *ii Also, solution lying in the stable manifold satisfy $|x_s(t) - y_s(t)| = O(\epsilon)$, if $|x_s(0) - y_s(0)| = O(\epsilon)$ for all times. If the solution lies in unstable manifold, similar result applies for the time interval $t \in [0, \epsilon)$.*

Proof of this theorem can be found in [26] but the conclusion (ii) says that this theorem can be applied to approximate stable (or unstable) manifolds in bounded sets and for studying

Poincaré section of the system of interest.

In the application of the averaging method, there are some important issues that should be controlled as mentioned in [26]. While interpreting the averaged system, one must give importance to the following point: global behavior of the original system may not be observed in the averaged system directly since n-dimensional maps and vector-fields do not display exactly the same behavior. Fortunately, under some conditions given in [27], certain global features in $\bar{f}$ display the same global features of f.

2.3.1. Comparison of Method of Averaging and Poincaré Map (Section) Analysis

The method of averaging and Poincaré maps are two different methods to study the time evolution of a dynamical system. While Poincaré map (section) analysis is based on the numerical integration of the system of interest, the method of averaging follows an analytical approach which continues with integration over a period. Although, both of these methods are useful tools to examine dynamical systems both have their own limitations. Since Poincaré section analysis depends on numerical integration, the result of this analysis is limited by the accuracy of the integrator which is used in simulation and this becomes important in long time interval simulation of the system. On the other hand, the method of averaging depends on perturbation theory and integration over a period which is based on making series expansions which are truncated to a certain order. Because of this truncation, errors are introduced so the averaged system can potentially display different behavior than the original one.

In many studies one of these two methods are used, however combination of both of these methods can give more accurate results according to [28] whose proof can be found in [26]. To claim that averaged system is meaningful to study, the original and averaged systems should display isomorphic phase space topologies (and fixed points) as their Poincaré sections. This can be said as required condition for the method of averaging.

2.4. LIE TRANSFORM

Lie transform method is special perturbation method developed for Hamiltonian systems. The main trick of the method is finding near-identity transformation which give simpler form of the Hamiltonian.

The procedure of the method as follows: Start with Hamiltonian system H of the form:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \quad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} \tag{2.16}$$

Expand the $H(q_i, p_i)$ in powers of the perturbation term ϵ:

$$H = H_o(q_i, p_i) + \epsilon H_1(q_i, p_i) + \epsilon^2 H_2(q_i, p_i) + \dots \tag{2.17}$$

After expansion apply canonical near-identity transformations from old coordinates to new coordinates $(q_i, p_i) \rightarrow (Q_i, P_i)$:

$$\begin{aligned} q_i &= Q_i + \epsilon F_{1i}(Q_i, P_i) + \epsilon^2 F_{2i}(Q_i, P_i) + \dots \\ p_i &= P_i + \epsilon G_{1i}(Q_i, P_i) + \epsilon^2 G_{2i}(Q_i, P_i) + \dots \end{aligned} \tag{2.18}$$

in Eq.2.18, F_{ji} and G_{ji} must be selected in a way that Hamiltonian equations are the same and we have canonical transformation. Combination of Eq. 2.17 and Eq.2.18 gives our new Hamiltonian K:

$$K(Q_i, P_i, \epsilon) = H(Q_i + \epsilon F_{1i}(Q_i, P_i) + \dots, P_i + \epsilon G_{1i}(Q_i, P_i) + \dots, \epsilon) \tag{2.19}$$

Eq.2.19 called as Kamiltonian, transformed perturbed Hamiltonian, in literature, thanks to Goldstein. Eq.2.19 can be expanded again into power series of ϵ:

$$K = K_0(Q_i, P_i) + \epsilon K_1(Q_i, P_i) + \epsilon^2 K_2(Q_i, P_i) + \dots \tag{2.20}$$

Important point in this argument is that $H_0 = K_0$, because of the fact that transformation is the near-identity one at $\epsilon = 0$. In addition to this, functions K_i based on function H_i where functions F_{ji} and G_{ji} will decide the transformation. Then , the idea behind this method comes into play: choose a transformation which keeps functions K_i simple.At this step, we need to understand and find a systematic way of defining identity transformation in subsection 2.4.1.

2.4.1. The Near-Identity Transformation

As it is understood main idea of the Lie transform method comes from identity-transformation. The relation between old and new variables is a function of ϵ. In this sense, ϵ plays a role like a time.

consider the evolution of transformation like as an evolution of Hamiltonian H:

$$\frac{dq_i}{d\epsilon} = \frac{\partial W}{\partial p_i} \quad , \frac{dp_i}{d\epsilon} = -\frac{\partial W}{\partial q_i} \tag{2.21}$$

where W is generating function and initial conditions are

$$\epsilon = 0, \quad q_i = Q_i, \quad p_i = P_i \tag{2.22}$$

As it can be seen W is function of ϵ,Q_i and P_i. Then expansion of W in power series of ϵ yields:

$$W = W_1(Q_i, P_i) + \epsilon W_2(Q_i, P_i) + \dots \tag{2.23}$$

At this step, it seen that the transformation required for Lie method (Eq. 2.18) can be found using Eq. 2.4.1 and Eq. 2.21 after writing p_i and q_i in Taylor series of ϵ:

$$\begin{aligned} q_i(\epsilon) &= q_i(0) + \epsilon \left.\frac{dq_i}{d\epsilon}\right|_{\epsilon=0} + \epsilon^2 \left.\frac{d^2q_i}{d\epsilon^2}\right|_{\epsilon=0} + \dots \\ p_i(\epsilon) &= p_i(0) + \epsilon \left.\frac{dp_i}{d\epsilon}\right|_{\epsilon=0} + \epsilon^2 \left.\frac{d^2p_i}{d\epsilon^2}\right|_{\epsilon=0} + \dots \end{aligned} \tag{2.24}$$

When we substitute Eq. 2.23 into Eq.2.21, we have:

$$\left.\frac{dq_i}{d\epsilon}\right|_{\epsilon=0} = \left.\frac{\partial W}{\partial p_i}\right|_{\epsilon=0} = \left.\frac{\partial W_1}{\partial p_i}\right|_{\epsilon=0} = \frac{\partial W_1}{\partial P_i} \tag{2.25}$$

remember that when $\epsilon = 0$, $p_i = P_i$. Finally, the near-identity transformation can be written as:

$$q_i = Q_i + \epsilon\frac{\partial W_1}{\partial P_i} + O(\epsilon^2), \quad p_i = P_i - \epsilon\frac{\partial W_1}{\partial Q_i} + O(\epsilon^2) \tag{2.26}$$

Note that in this derivation,we derive near-identity transformation formulas for order ϵ. Higher order derivation can be done using the same methodology. For much more details of Lie transform method, one can refer to [29, 30].

In this study, we applied Poincaré section to investigate the possible parameter regions where system make transition from ordered behavior to indeterministic nature then we make use of the perturbation methods namely averaging method and Lie transform on the Matinyan-Yang-Mills-Higgs (MYMH) system which is a Hamiltonian system. In chapter 3, all the analysis results are presented, and observations are discussed with details.

3. ANALYSIS OF NON-INTEGRABLE MATINYAN-YANG-MILLS-HIGGS AND SIMULATION OF DYNAMICS OF THE SYSTEM

In this chapter, by making use of the theoretical information and tools in Chapter 2, we make the analysis of the Matinyan-Yang-Mills-Higgs (MYMH) System. The MYMH system has been used for modelling the onset of chaotic behavior in the classical Yang Mills system [31]. It turns out that limited suppression of chaotic behavior is also exhibited. Analysis of the system is important since this Hamiltonian system can represent our recent understanding on the stability of the universe and the mechanism for the onset of instability.

This chapter is organized as follows: In section 3.1, we introduce the MYHM system and we make linear stability analysis of it. After stability analysis, by playing with parameters of the system,we try to investigate the region that system displays chaotic behavior and possible parameter regions where the system makes transition from order to chaos or vice versa by Poincare Map analysis with numerical simulations. Then in section 3.2, we try to find approximately conserved quantity through some algebraic manipulations and we analyze the integrability of the system under certain conditions. Finally in sections 3.3 and 3.4, we make use of Lie transform and averaging methods on our system to investigate possible bifurcation scenarios.

3.1. MYMH HAMILTONIAN AND IT'S LINEAR STABILITY ANALYSIS

MYMH Hamiltonian, in two-dimensional Cartesian coordinates, is described by the following Hamiltonian H:

$$H = \frac{p_x^2 + p_y^2}{2} + \frac{g^2(x^2+y^2)}{2} + \frac{x^2y^2}{2} + V_H(y) \tag{3.1}$$

where $V_H(y)$:

$$V_H(y) = -\frac{y^2}{2} + \frac{ay^4}{4} \tag{3.2}$$

Where **g** and **a** are real constant parameters and x, y and p_x, p_y are generalized coordinates and corresponding momenta respectively. The term $\frac{g^2}{2}(x^2+y^2)$ is the harmonic oscillatory term which has the potential to have a stabilizing effect on the system and lead the system to a limit cycle like structure. The second important term is the Higgs term: $\frac{1}{4}ay^4$. In a detailed analysis of the Higgs term, discussed very intuitively in [31], is found that for high energy values it leads the system to display chaotic characteristics. It is also important to note that there is no sharp transition from chaotic regions to ordered regions by small changes in energy values.

Before, starting analysis of the system we, rewrite the system (3.1) in the following form:

$$H = \frac{p_x^2+p_y^2}{2} + \frac{(w_1^2x^2+w_2^2y^2)}{2} + \frac{x^2y^2}{2} + \frac{ay^4}{4} \tag{3.3}$$

where $w_1^2 = g^2$ and $w_2^2 = g^2 - 1$.

Hamiltons equations of motion for H(Eq. 3.3) are:

$$\frac{dp_x}{dt} = -w_1^2x - xy^2 \tag{3.4}$$

$$\frac{dp_y}{dt} = -ay^3 - w_2^2y - x^2y \tag{3.5}$$

$$\frac{dx}{dt} = p_x \tag{3.6}$$

$$\frac{dy}{dt} = p_y \tag{3.7}$$

The trivial fixed point of the system is the origin. The other fixed points of the systems are:

(i) $x = \pm\sqrt{aw_1^2 - w_2^2}$, $y = \pm iw_1$, $p_x = 0$, $p_y = 0$
(ii) $x = 0$, $y = \pm\frac{iw_2}{\sqrt{a}}$, $p_x = 0$, $p_y = 0$

The fixed point(s) in (i) can only take real value for $w_1 = w_2 = 0$, which is the origin. The

second fixed point(s) can take real values for the cases where $a < 0$ and $w_2 \geq 0$. To classify the fixed points, we firstly calculated the Jacobian matrix J:

$$J = \begin{pmatrix} 0 & 0 & -w_1^2 - y^2 & -2xy \\ 0 & 0 & -2xy & -3ay^2 - w_2^2 - x^2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{3.8}$$

Firstly, to look at the stability of origin, we find the characteristic equation of J at the origin:

$$det(J - \lambda I) = \lambda^4 - (w_2^2 - w_1^2)\lambda^2 - w_1^2 w_2^2 \tag{3.9}$$

We find the following eigenvalues λ_i:

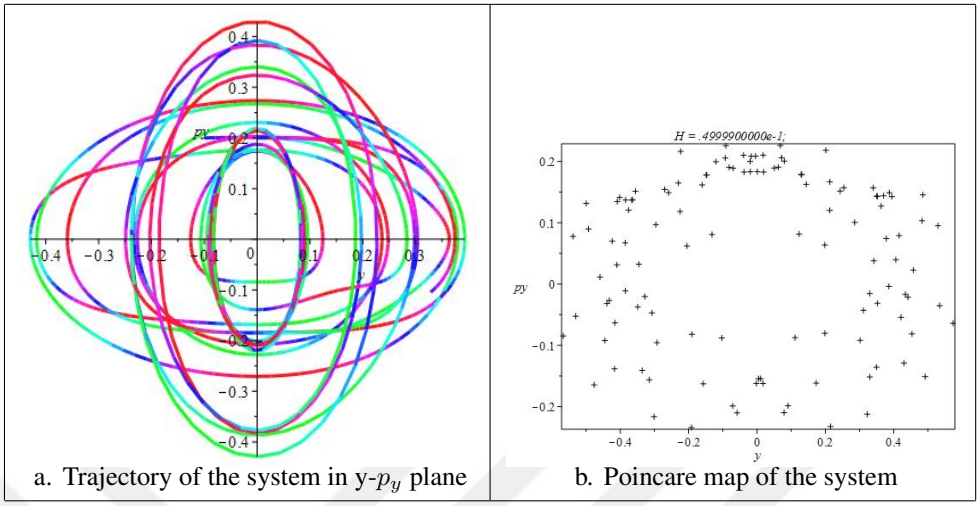
$$\lambda_{1,2} = \pm|w_1|i, \quad \lambda_{3,4} = \pm w_2 \tag{3.10}$$

The eigenvalues of MYMH system are symmetric about both real and imaginary axis. To claim that the origin is stable all according to Eq. 3.9 eigenvalues must lie on the imaginary axis. While analyzing state of system like in Eq. 3.3, the important observation can be made in case $w_1/w_2 << 1$. For the internal resonance case the relation like the following should be satisfied:

$$k_1 w_1 + k_2 w_2 = 0 \tag{3.11}$$

where k_1 and k_2 are integers. Firstly, we investigated case: $w_1/w_2 = 1/2$ in other words 1:2 resonance. In this situation, periodic set of solutions curves exist around the origin, and in that case, we observe limit cycles. In Fig. 3.3, the situation, where $w_1/w_2 = 1/2$, is illustrated. For the cases $w_2 \neq 0$, the origin is center, since there exist one positive eigenvalues of J(see Fig. 3.1). Interesting situation may happen where $w_1 = 0$ and $w_2 \neq 0$, where we have double zero eigenvalues which will result in bifurcation. In this situation the energy transfer from high frequency mode to low frequency mode may happen. To analyze this kind of situation, perturbation methods are used. In the next two subsections 3.3 and 3.4,

result of applying perturbation methods to MYMH(Eq. 3.3) are discussed.



a. Trajectory of the system in y-p_y plane b. Poincare map of the system

Figure 3.1. Trajectory and Poincare map of the MYMH system with initial conditions p_x=0.2440286868, p_y=0.2, x= 0.0, y=-0.1 (where $w_1 = 0.2, w_2 = 0.2, a = 1.0$)

Secondly, we analyze the stability of the fixed point (ii) in the case that $w_2 \geq 0$ and $a < 0$. Using the Jacobian J(Eq. 3.8),we find the characteristic equation:

$$a^2\lambda^4 + (a^2w_1^2 - 2a^2w_2^2 - aw_2^2)\lambda^2 + 2aw_2^2(w_2^2 - aw_1^2) = 0 \tag{3.12}$$

Then we find the eigenvalues of Eq. 3.12 for $w_2 = 0$ and $a < 0$. We obtain the following eigenvalues:

$$\lambda_{1,2} = 0 \quad \text{and} \quad \lambda_{3,4} = \pm w_1 i \tag{3.13}$$

This situation is the same as in the case where Eq. 3.10

Up to now, we have studied the MYMH system (Eq. 3.3) by linear stability analysis. To understand the role of the parameters g and a on dynamics of system (Eq. 3.1), we numerically search for range of parameters where the invariant torus is distorted and eventually chaotic nature of the system is apparent.

Firstly, we study the effect of parameter g on the dynamics of the system.To view the role

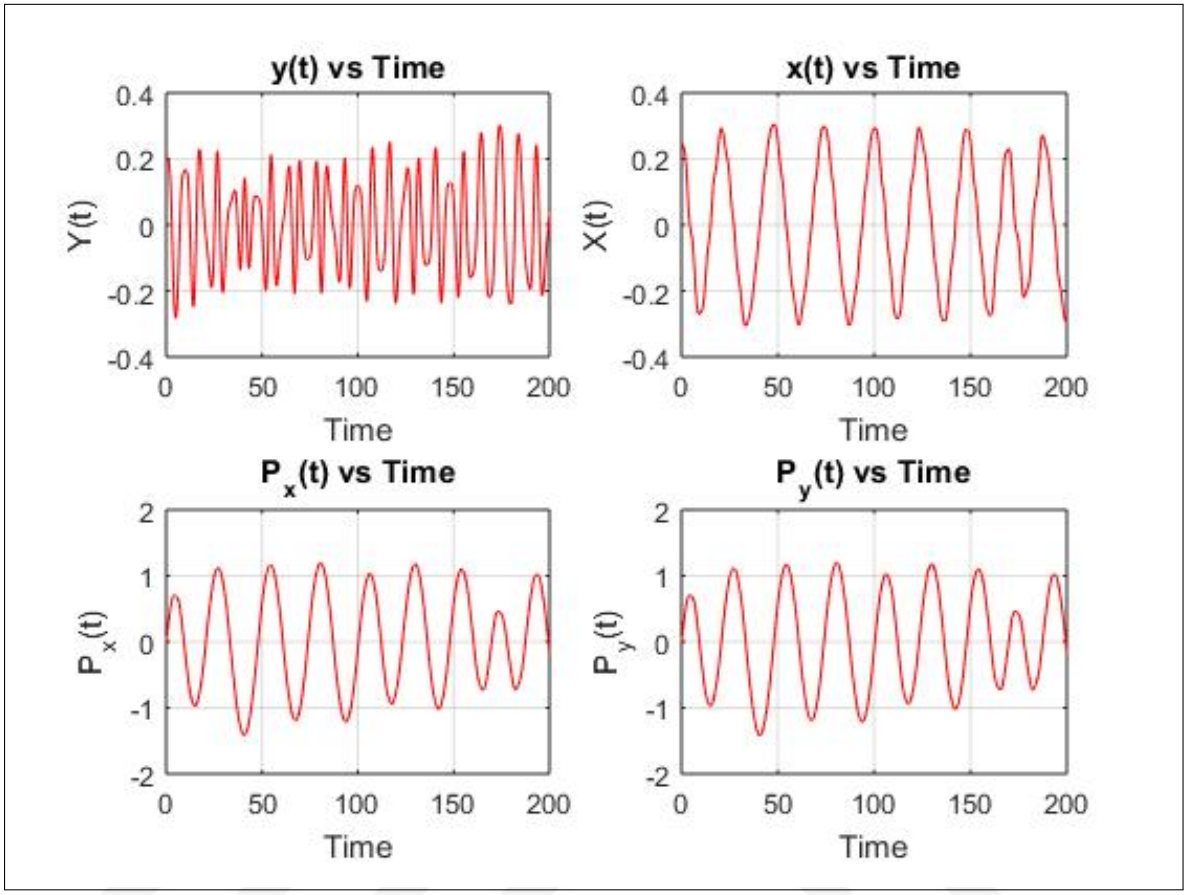
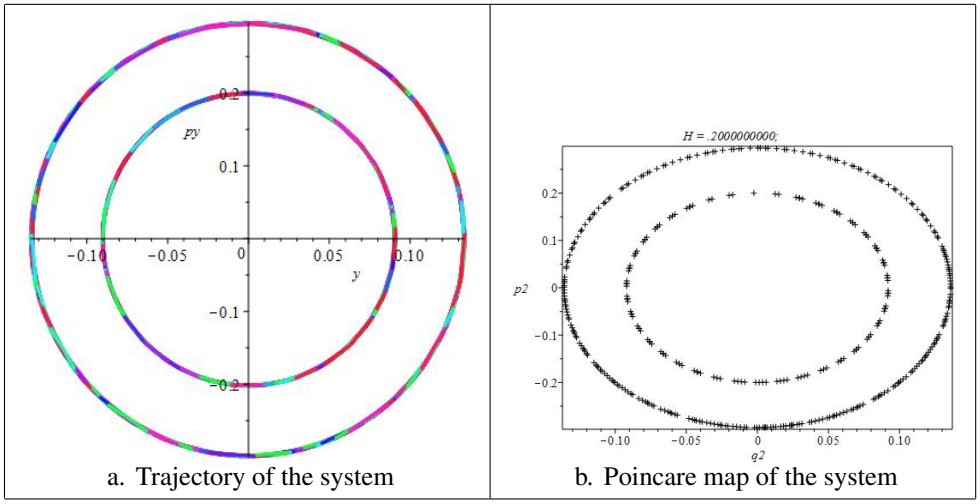


Figure 3.2. Trajectory of the MYHM system(Eq.3.3) for 1:1 resonance case

of g on stability of the system,we use Poincare map of the system by setting $x = 0$ and plot Poincare sections on $(y - p_y)$ plane.

When, Fig. 3.5.a is the Poincare map of the system, in the absence of g (g=0) and it shows that the fixed point $(0, 0, 0, 0)$ behaves as a source like point and there exist two closed curves so that motion is irregular. When we increase the value of parameter g from 0 to 0.5 (Fig. 3.5.c, the behavior of the system changes and we observe a closed orbit around origin in the Poincaré section. For higher values of the parameter g the system reaches structurally stable cycle of limited spatial extent, which can be seen in the Fig. 3.5.e and Fig. 3.5.f. In the light of these results from Poincare section analysis of system, parameter g can induce a significant changes in the dynamical behavior of the system in such a way that system makes transition from one chaotic behavior type to another one. Note that for each simulation result, we have checked that energy is conserved.



a. Trajectory of the system b. Poincare map of the system

Figure 3.3. Trajectory and Poincare map of the MYMH system where $w_1/w_2 = 1/2$ and $H = 1/5$

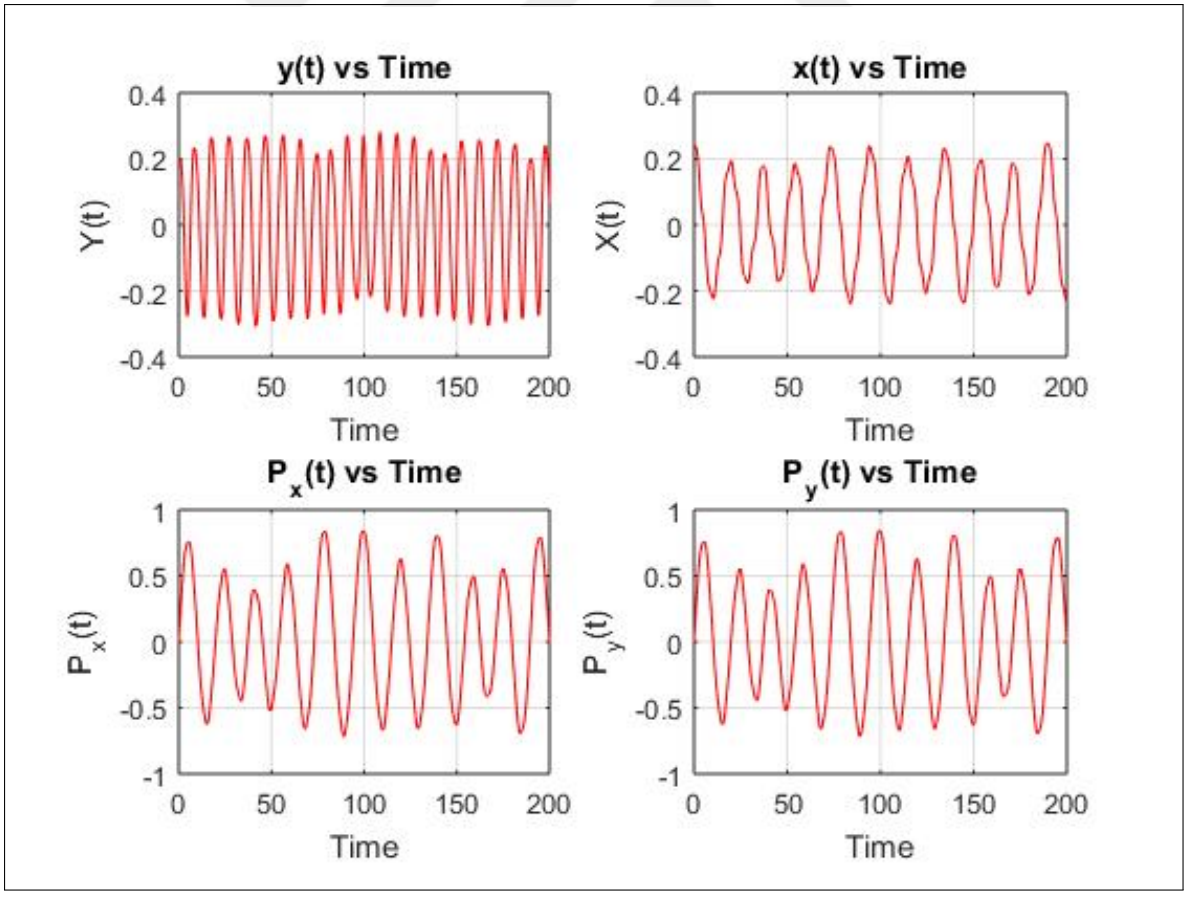


Figure 3.4. Trajectory of the MYMH system (Eq.3.3) for 1:2 resonance case

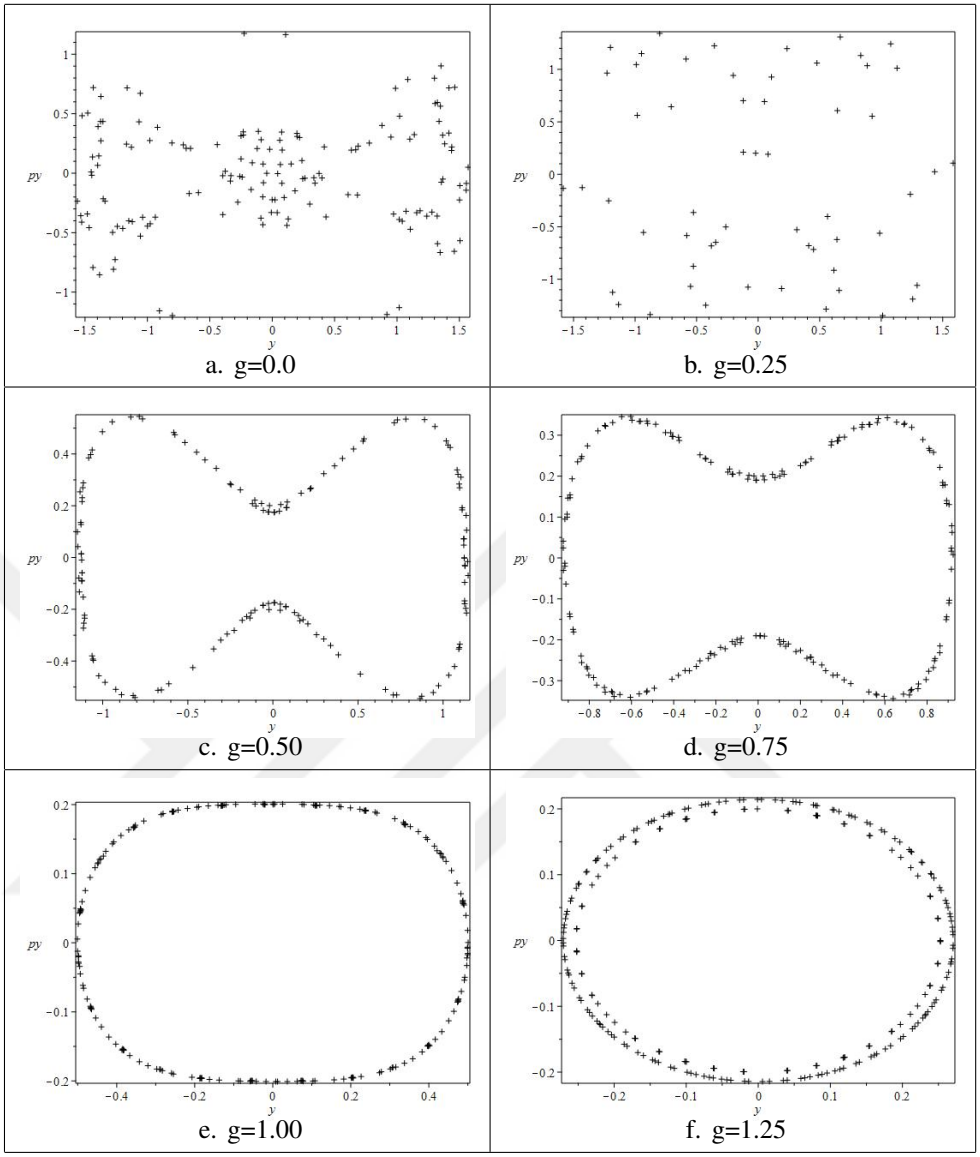


Figure 3.5. Poincare Maps of MYMH system at $E = 0.05$

Up to this point, we have said that for low energy value system behaves chaotically by looking at Poincare sections. To check this result, we have looked at Lyapunov exponent spectrum of the system (Fig. 3.6) and their corresponding largest Lyapunov exponents for each values of parameter g given by Wolf algorithm in Table 3.1.

Table 3.1. Lyapunov exponents of the system for different values of parameter g.

Parameter	**Maximal Lyapunov exponents**			
g	λ_1	λ_2	λ_3	λ_4
0.0	0.0015	0.0006	-0.0006	-0.0015
0.25	0.0011	0.00035	-0.00035	-0.0011
0.50	0.0012	0.00039	-0.00039	-0.0012
0.75	0.0018	0.0015	-0.0015	-0.0018
1.00	0.0030	0.0026	-0.0026	-0.0030
1.25	0.0055	0.0013	-0.0013	-0.0055

In Fig. 3.6, Lyapunov exponent spectrum of system for $g = 0.0$ and $g = 0.25$ is given and in both cases, there exists 2 positive, 2 negative exponents and sum of all the exponents is equal to 0. In addition to this, since there are two positive exponents exist, chaos can be observed for these values of parameter g. Furthermore, for increasing values of parameter g, largest Lyapunov exponents become close to zero so that system is going to be nearly in limit cycle.

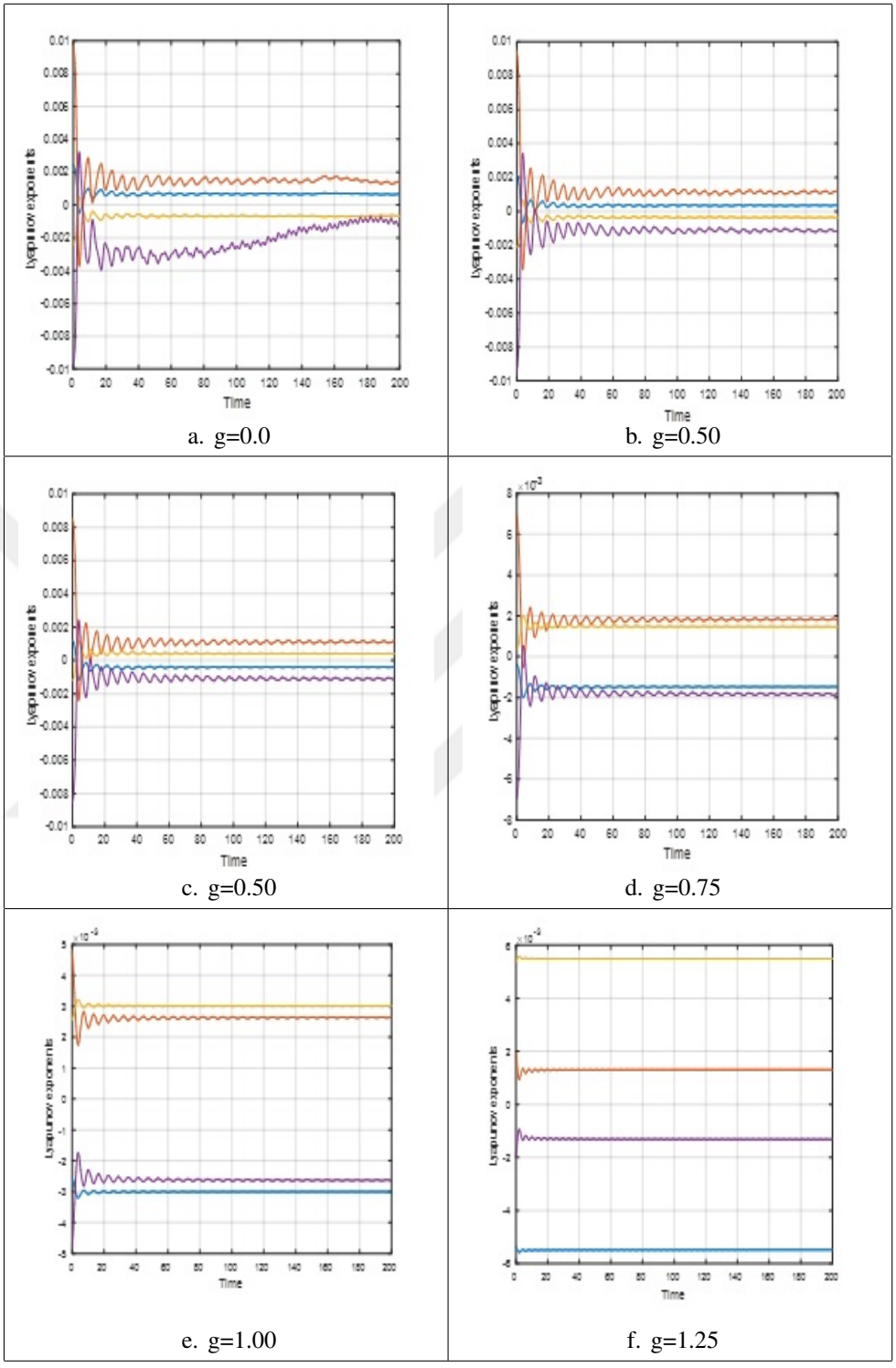


Figure 3.6. Lyapunov exponent spectrum of the system for different values of parameter g

3.2. INTEGRABILITY OF MYMH

In this section, we consider the problem of finding approximately conserved quantities of MYMH system (Eq.3.1) given in section 3.1. In Hamiltonian dynamics, if there exists a second integral of motion I_2 other than energy H itself,the Poisson bracket(PB) of I_2 with H denoted as I_2, H, vanishes. The following two terms $(x^2 + y^2)$ and $(xp_y - yp_x)$ repeatedly appear in the Toda truncations [32, 33]. We will see that $(xp_y - yp_x)$ will be immaterial in the MYMH case, since it depends on rotational invariance in $x - y$. Similar results covering the existence of approximate integral in Toda case ,for specific values of parameters can be found in[34]. Many terms in the MYMH system resemble these of the Toda truncations.

In order to generalize this to the MYMH system, the following candidate third integral is constructed:

$$\frac{1}{2}(p_x^2 + p_y^2) + \frac{c_1}{2}(x^2 + y^2) + c_3(p_y x - p_x y) + (c_4 x + c_5 y)(x^2 + y^2) \tag{3.14}$$

where
$2c_2 = c_1, 4c_1 = g^2, g^2 = 1, c_3 = 0$. To find possible candidate for the I_2 we follow algebraic manipulation by making partial sums for truncated MYMH Hamiltonian to successive homogeneous orders and the analytic isolating integral given by I_{ij}.

$$\begin{aligned} H_T(1) &= 0 \\ H_T(2) &= \tfrac{p_x^2+p_y^2}{2} + \tfrac{g^2(x^2+y^2)}{2} - \tfrac{y^2}{2} \\ H_T(3) &= 0 \\ H_T(4) &= \tfrac{x^2y^2}{2} + \tfrac{py^4}{4} \end{aligned} \tag{3.15}$$

Possible candidates for I_{ii} are given in Eq 3.16.

$$\begin{aligned} &I_{ii}(1) = 0 \\ &I_{ii}(2) = (p_x y - x p_y) \\ &I_{ii}(3) = 4(3p_x x^2 - 6p_y^2 p_x - 6xyp_y + 2p_x^3 - p_x y) \\ &I_{ii}(4) = 24(p_x y - x p_y)(x^2 + y^2) \end{aligned} \tag{3.16}$$

The non-vanishing terms of PBs' of $\{H_T(i), I_{ii}(i)\}$ are:

$$\{H_T(2), I_{ii}(2)\} = xy \tag{3.17a}$$

$$\{H_T(2), I_{ii}(3)\} = 4(-6g^2p_y^2y + 6p_y^2x - 12g^2p_xp_yy + 18p_xp_yy + p_xp_y + 6g^2p_x^2x \tag{3.17b}$$

$$-6p_x^2x - 6g^2xy^2 + 6xy^2 - g^2xy + 3g^2x^3)$$

$$\{H_T(2), I_{ii}(4)\} = 24(2xyp_y^2 - 2p_xp_yy^2 + 2p_xp_yx^2 - 2 - 2p_x^2xy + xy^3 + x^3y) \tag{3.17c}$$

$$\{H_T(4), I_{ii}(2)\} = xy(-ay^2 + y^2 - x^2) \tag{3.17d}$$

$$\{H_T(4), I_{ii}(3)\} = 4y(-6xyp_y^2 - 12ap_xp_yy^2 - 12p_xp_yx^2 + 6xyp_x^2 - 6axp_y^3 \tag{3.17e}$$

$$-xy^2 - 3x^3y)$$

$$\{H_T(4), I_{ii}(4)\} = 24xy(-ay^4 + y^4 - ax^2y^2 - x^4) \tag{3.17f}$$

Then, the partial sums of $H_T(j)$ and $I_{ii}(j)$ are defined as follows:

$$I_{iss}(n) = \sum_{j=1}^{n} I_{ii}(j) \tag{3.18a}$$

$$H_{Ts}(n) = \sum_{j=1}^{n} H_{Ts}(n) \tag{3.18b}$$

using the definitions in Eq.3.18, the PB of the partial sums of the list elements of truncated

Toda Hamiltonian H_T and the isolating integrals I_{ii} is defined as follows:

$$\{H_{Ts}(n); I_{iis}(n)\} \tag{3.19}$$

where $n = 1, \ldots, \infty$. According to the truncations of $H_T(j)$ and $I_{ii}(j)$ given above, the PBs of the partial sums of equal terms up to fifth order truncations are obtained as

$$\{H_{Ts}(1), I_{iis}(1)\} = 0 \tag{3.20a}$$

$$\{H_{Ts}(2), I_{iis}(2)\} = xy \tag{3.20b}$$

$$\{H_{Ts}(3), I_{iis}(3)\} = -24p_y^2x(g^2-1) + p_xp_yy(72-48g^2) + 4p_xp_y + 12g^2x^3 \tag{3.20c}$$
$$+24p_x^2x(g^2-1) + 24xy^2(1-g^2) - xy(1-4g^2)$$

$$\{H_{Ts}(4), I_{iis}(4)\} = 2(-12p_y^2y^2x + 24p_y^2xy - 6p_y^2x^3 - 12gp_y^2x - 24bp_xp_yy^3 \tag{3.20d}$$
$$-24p_xp_yy^2) + O(5)terms$$

According to Eq.3.20d, the Poisson bracket of fourth order truncations give fifth order terms which leads the idea of possible conserved quantity candidates.Yet, if PB of n^{th} order truncations is of higher order, the indicated quantity is considered to be conserved to order n.

3.3. ANALYSIS OF MYMH SYSTEM USING LIE TRANSFORM

To analyze the system (3.1) using Lie transform method, we rewrite the system the following form:

$$H = \epsilon \left(\frac{a\, y^4}{4} + x^2\, y^2 \right) + \frac{{w_2}^2 y^2 + {w_1}^2 x^2}{2} + \frac{{p_y}^2 + {p_x}^2}{2} \tag{3.21}$$

where ϵ denotes the perturbation constant and note that in this representation,we absorb the $-y^2/2$ term into constant w_2. In this representation, the term $x^2y^2/2$ is treated as a perturbation term, which leads to chaotic behavior on the system during the absence of harmonic oscillatory term $g^2(x^2+y^2)/2$ in Eq. 3.1.

Firstly, we apply transformation from Cartesian coordinates to action-angle variables $(x, y, p_x, p_y) \rightarrow (J_1, \phi_1, J_2, \phi_2)$ using:

$$x = \sqrt{\frac{2J_1}{w_1}} \sin(\phi_1) \quad p_x = \sqrt{2J_1 w_1} \cos(\phi_1) \tag{3.22}$$

$$y = \sqrt{\frac{2J_2}{w_2}} \sin(\phi_2) \quad p_y = \sqrt{2J_2 w_2} \cos(\phi_2) \tag{3.23}$$

where J_i's are action and ϕ_i's are angle variables. After the transformation, Eq. 3.23 takes the following form:

$$H = J_2\, w_2 + \frac{J_1\, J_2 \cos{(2\phi_2 + 2\phi_1)\epsilon}}{2w_1 w_2} + \frac{J_1\, J_2 \cos{(2\phi_2 - 2\phi_1)\epsilon}}{2w_1 w_2} - \frac{J_1\, j_2 \cos{(2\phi_2)\epsilon}}{w_1 w_2} - \tag{3.24}$$

$$\frac{J_1\, J_2 \cos{(2\phi_1)\epsilon}}{w_1 w_2} + \frac{J_1\, J_2 \epsilon}{w_1 w_2} + \frac{J_2^2 \cos{(4\phi_2)a\epsilon}}{8{w_2}^2} - \frac{J_2^2 \cos{(2\phi_2)a\epsilon}}{2{w_2}^2} + \frac{3J_2^2 a\epsilon}{8{w_2}^2} + J_1\, w_1$$

Eq. 3.24 seems very complicated. The first order term in the power series expansion of

Kamiltonian K about perturbation parameter ϵ called K_1:

$$K_1 = -\frac{I_1\, I_2 \cos\left(2\psi_1\right)}{w_1 w_2} + \frac{I_1\, I_2}{w_1 w_2} + \frac{3 I_2^2 a}{8 {w_2}^2} \tag{3.25}$$

The generating function W_1:

$$W_1 = \frac{\sin\left(2\psi_2\right)\left(2 I_1\, I_2\, w_2 + I_2^2 a\, w_1\right)}{4 w_1 {w_2}^3} - \frac{I_1\, I_2 \sin\left(2\psi_2 + 2\psi_1\right)}{4 w_1 {w_2}^2} + \sin\left(2\psi_1\right) B \tag{3.26}$$

$$- \frac{I_1\, I_2 \sin\left(2\psi_2 - 2\psi_1\right)}{4 w_1 {w_2}^2} - \frac{I_2^2 \sin\left(4\psi_2\right) a}{32 {w_2}^3} + \cos\left(2\psi_1\right) A$$

The self-similarity transformations from (J_i, ϕ_i) to (I_i, ψ_i) is done by:

$$J_1 = \epsilon \left(\frac{I_1 I_2 \sin(2\psi_1) cos(2\phi_2)}{w_1 w_2^2} + 2B \cos 2\psi_1 - 2A \sin(2\psi_1) \right) + I_1 \tag{3.27a}$$

$$\phi_1 = \epsilon \left(\frac{I_2 \sin\left(2\psi_2 + 2\psi_1\right)}{8 w_1 {w_2}^2} + \frac{I_2 \sin\left(2\psi_2 - 2\psi_1\right)}{8 w_1 {w_2}^2} - \frac{I_2 \sin\left(2\psi_2\right)}{4 w_1 {w_2}^2} \right) + \psi_1 \tag{3.27b}$$

$$J_2 = I_2 + \epsilon \left(\frac{\cos\left(2\psi_2\right)\left(2 I_1\, I_2\, w_2 + I_2^2 a\, w_1\right)}{2 w_1 {w_2}^3} - \frac{I_1\, I_2 \cos\left(2\psi_2 + 2\psi_1\right)}{2 w_1 {w_2}^2} \right. \tag{3.27c}$$

$$\left. - \frac{I_1\, I_2 \cos\left(2\psi_2 - 2\psi_1\right)}{2 w_1 {w_2}^2} - \frac{I_2^2 \cos\left(4\psi_2\right) a}{8 {w_2}^3} \right)$$

$$\phi_2 = \psi_2 + \epsilon \left(-\frac{\sin\left(2\psi_2\right)\left(2 I_1\, w_2 + 2 I_2 a\, w_1\right)}{4 w_1 {w_2}^3} + \frac{I_1 \sin\left(2\psi_2 + 2\psi_1\right)}{4 w_1 {w_2}^2} \right. \tag{3.27d}$$

$$\left. + \frac{I_1 \sin\left(2\psi_2 - 2\psi_1\right)}{4 w_1 {w_2}^2} + \frac{I_2 \sin\left(4\psi_2\right) a}{16 {w_2}^3} \right)$$

We choose *A* and *B* to simplify the K_1

$$A = 0, \text{ and } B = -\frac{I_1 I_2}{8 w_1 w_2^2} \tag{3.28}$$

The transformed perturbed Hamiltonian,called as Kamiltonian K is:

$$K = I_1\, w_1\, + I_2\, w_2 + \epsilon \left(-\frac{I_1\, I_2\, \cos\left(2\,\psi_1\right)}{2\, w_1\, w_2} + \frac{I_1\, I_2}{2\, w_1\, w_2} + \frac{3\, I_2^2\, a}{8\, w_2^2} \right) + O(\epsilon^2) \tag{3.29}$$

When you look at Kamiltonian K there is no-term related to angle coordinate ψ_2, so that I_2 is constant of motion (to $O(\epsilon^2)$). Apparently, K(Eq. 3.29) is also a constant of the motion in this autonomous two degree of freedom system.Thus it follows that $K - w_2 I_2$ is a constant of the motion, so that:

$$L = -I_2 \left(\frac{I_1 cos(2\psi_1)}{2 w_1 w_2} + \frac{I_1}{2 w_1 w_2} + \frac{3a}{8 w_2^2} \right) + O(\epsilon) = constant \tag{3.30}$$

and K_1 is a first integral to $O(\epsilon^2)$. We check the Poisson bracket of Eq. 3.30 with K(Eq. 2.19 to order ϵ^2:

$$\{K, L\} = 0 \tag{3.31}$$

So this result agrees with the observation that K_1 can be considered as approximate integral to our system.

3.4. ANALYSIS OF MYMH WITH METHOD OF AVERAGING

In this subsection, we analyze MYMH system(Eq. 3.3) by using the method of averaging, we analyze Eq. 3.3 for different $w_1 : w_2$ resonance cases. Firstly, we investigate the situation

where $w_1 = w_2 = w$. The canonical transformations applied to the MYHM system are:

$$\begin{aligned} x &= u\cos wt + v\sin wt \\ y &= -u\sin wt + v\cos wt \\ p_y &= p_u\cos wt + p_v\sin wt \\ p_x &= -p_u\sin wt + p_v\cos wt \end{aligned} \tag{3.32}$$

where u,v, p_u and p_v are functions of the time. Subsequently, we find the averaged system for the 1:1 resonance case by integrating the Hamilton canonical equations and averaging over the period $2\pi/w$(the period of the quadratic oscillator part). The resulting averaged equations are:

$$\begin{aligned} \dot{u} &= p_u - wv \\ \dot{v} &= p_v + wu \\ \dot{p_u} &= \tfrac{-3au^3-3auv^2-8wp_v-2u^3-2uv^2-8w^2u+4u}{8} \\ \dot{p_v} &= \tfrac{-3av^3-3au^2v+8wp_u-2v^3-2u^2v-8w^2v+4v}{8} \end{aligned} \tag{3.33}$$

There exist at equilibrium two complex conjugate roots and one real fixed point $(u, v, p_u, p_v) = (0, 0, 0, 0)$ of the system. It is also important to note that the same real fixed point $(u, v, p_u, p_v) = (0, 0, 0, 0)$ also exists in the original system so that the Poincare section analysis of original system can give correct approximation to the original system by satisfying necessary condition mentioned in the subsection 3.1.

To check the linearized stability at the real fixed point of the averaged system(Eq. 3.33), we calculate the Jacobian matrix of the linear part of the averaged system there.

$$J_{(0,0,0,0)} = \begin{pmatrix} 0 & -w & 1 & 0 \\ w & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \end{pmatrix} \tag{3.34}$$

To find the eigenvalues λ_i of J, we find the characteristic equation:

$$\det|J - \lambda I| = 4\lambda^4 + 4(w^2 - 1)\lambda^2 + 1 = 0 \tag{3.35}$$

The corresponding eigenvalues are:

$$\lambda_{1,2} = \frac{\pm i\sqrt{w^2-2}+w}{2} \quad \text{and } \lambda_{3,4} = \frac{\pm i\sqrt{w^2-2}\mp iw}{2} \tag{3.36}$$

When eigenvalues given in Eq. 3.36 is taking into consideration, stability of the origin for the averaged system(Eq. 3.33) depends on the choice of w.When $w^2 = 2$, we have two complex conjugate purely imaginary eigenvalues and two real eigenvalues and this result suggests that there is a Hopf bifurcation change according to the value of the parameter(driving frequency) ω. We reach this conclusion because of the following reason: Hopf bifurcation typically occurs when a complex conjugate pair of eigenvalues of the linearized flow at a fixed point becomes purely imaginary.

To be more accurate about the role of the constant g in the stability of the system, we write the following Lyapunov function $V(u, v, p_u, p_v)$ by looking at the following observation for high values of g where g^2 is much greater than 1:

$$p_u\dot{p_u} + p_v\dot{p_v} = -g^2(up_u + vp_v) \tag{3.37}$$

Then we construct the following positive definite Lyapunov function $V(u, v, p_u, p_v) > 0$:

$$V = g^2(u^2 + v^2) + p_u^2 + p_v^2 \tag{3.38}$$

V is positive definite for all points other than the origin of the averaged system for the given g^2 limit above. We also compute the derivative of Lyapunov function $\dot{V}$:

$$\dot{V}(u, v, p_u, p_v) = 0 \tag{3.39}$$

Since the time derivative of the Lyapunov function V equals to 0, according to LaSalles' theorem [35], there exists an invariant set M where every motion starting in converges either to the origin or to the limit cycle. Furthermore, it might be said that the Lyapunov function that we offer can be considered as a generalized energy function. To check validity of this assumption, we have simulated the averaged system for g=10 ($g^2 >> 1$) and get the

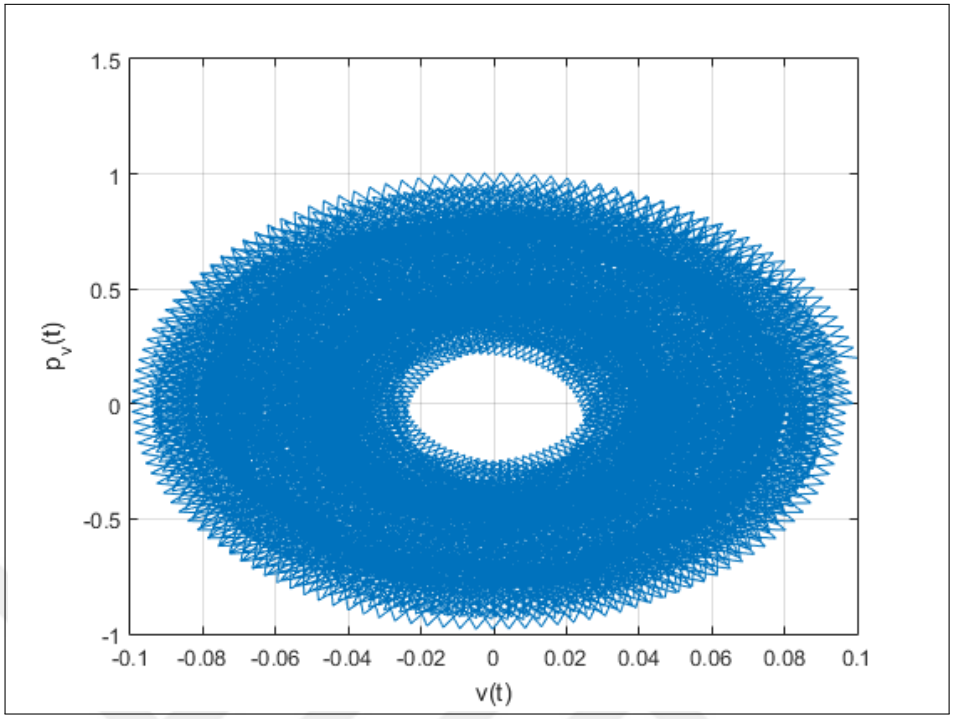


Figure 3.7. Trajectory of linearized averaged system for $g = 10$

following Poincare section shown in Fig. 3.7. Then, we have used Poincare section analysis to understand the time evolution of the averaged linearized system. We simulated the system for a large range values of g but, we observed that the fixed point (0,0,0,0) behaves like an attractor especially for values $g \geq 0.75$.In Fig. 3.8, time evolution of each components of the system is presented. In Fig. 3.9 $u(t)$ vs $v(t)$ and $p_v(t)$ vs. $v(t)$ plots are given and according to these graphs, there exists an unstable torus around the origin.

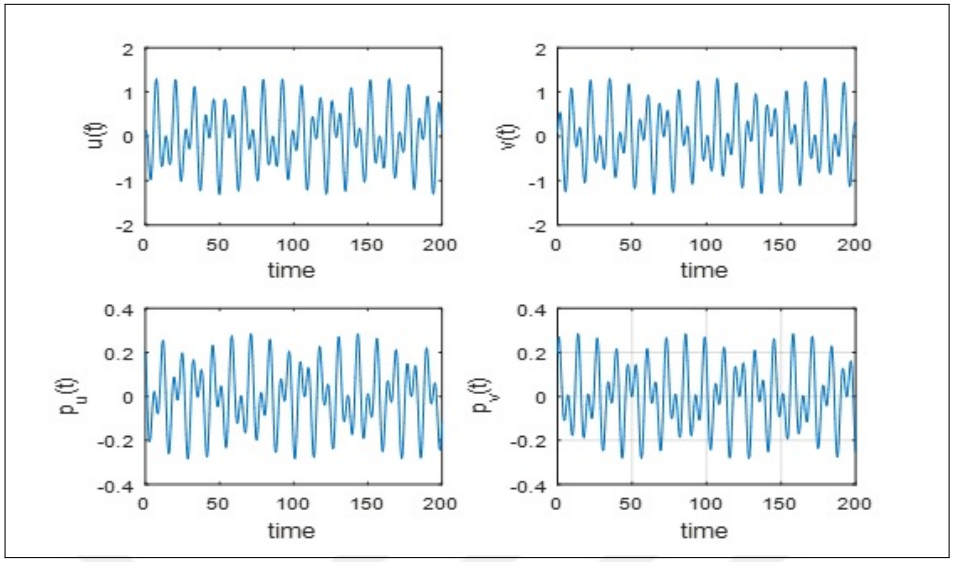
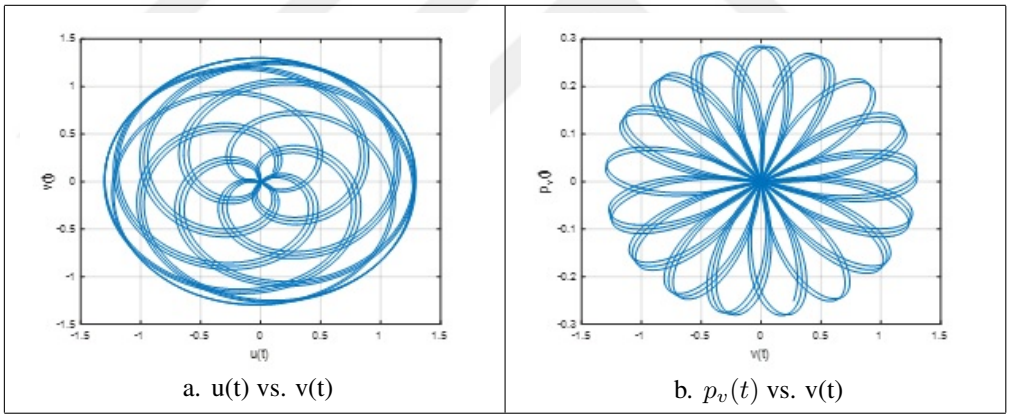


Figure 3.8. Time evolution of linearized averaged system ($g = 0.74$)



a. u(t) vs. v(t)

b. $p_v(t)$ vs. v(t)

Figure 3.9. Time evolution of linearized averaged system ($g = 0.74$) in $u(t)$ vs $v(t)$ and $p_v(t)$ vs. $v(t)$ spaces

For smaller values of parameter g, the system displays regular behavior and p_v and v have a linear relationship.

4. NEURAL NETWORKS AND THEIR USAGE IN CHAOTIC DYNAMICAL SYSTEMS

In many scientific study fields, estimation of future values of a system in interest is very important in understanding the dynamical evolution. The physical systems, which are governed by a set of equations of motion, are commonly investigated by integrating them forward in time and by this way, evolution of the system is studied. In chaos theory, for a given time series, particularly with a series originating form a dynamical system, it is important to understand whether time series is chaotic or not which means that there is a sensitive dependence on initial conditions or not. One the most commonly accepted parameter to talk about existence of chaos in given time series is its Lyapunov exponent which gives the rate of convergence (or divergence) of nearby trajectories in state space. For a time, series, positive Lyapunov exponent is indicator of chaos. In addition to this, Lyapunov exponents are used to calculate important measures of a dynamical system. Dimension of the attractor is calculated by Kaplan-Yorke conjecture using Lyapunov exponents and Kolmogorov-Sinai entropy can also be calculated by making use of them.

However, many cases, governing equations of motion for a given time series are unknown, such as stock market indices in economics or experimental data. To forecast chaotic time series, there are many methods other than neural networks such as Taylor series expansion, radial basis functions, nonparametric kernel regressions. These methods are based on interpolation and approximation of unknown function by use of scattered data points. In Taylor series expansion approach the main disadvantage is rapidly increasing order of expansion and according to Casdagli [36], there is no guaranteed order of convergence for dimensions $n > 1$, and higher polynomial degrees have a wide oscillation tendency. Nonparametric kernel regression is a method that depends on estimating probability density function from observed time series, but it has some drawbacks mentioned in [22]. On the other hand, in the literature, there are many studies which demonstrate the capabilities of neural networks in forecasting chaotic time series. For example, two-layered feedforward neural networks are used in [37], it gives promising result in estimation of chaotic time series generated from Lorenz system, Henon and Logistic maps. In another study [20],

Gencay demonstrates that nonlinear noisy time series can be modeled quite accurately by single-layer feedforward neural networks.

In this chapter, we use three different neural network architectures to test capabilities of neural networks in forecasting time series generated from different dynamical systems. In addition to forecasting time series, using the feedforward neural network with single hidden layer, Lyapunov exponents of the studied systems are forecast. This chapter is organized as follows: in section 4.1, we define the fundamentals of neural network and their working mechanism. In section 4.2, we present the performance of each network in terms of forecasting ability. In section 4.3, an algorithm designed to calculate using the neural network is presented and evaluating performance of each network in estimation of Lyapunov exponent is reported.

4.1. NEURAL NETWORK AND LEARNING MECHANISM

Neural Networks are the mathematical model which tries to imitate the working mechanism of neurons in our brain. Earliest form of neural networks, called **perceptrons** were developed in the 1950s and 1960s by the scientist Frank Rosenblatt. Basically, neural networks are designed to imitate the working mechanism of neurons in brain. They are very commonly used tools in many science fields: such as image recognition, classification, time series forecasting, pattern recognition etc... Yet, they do not completely share the same mechanism with biological neurons.

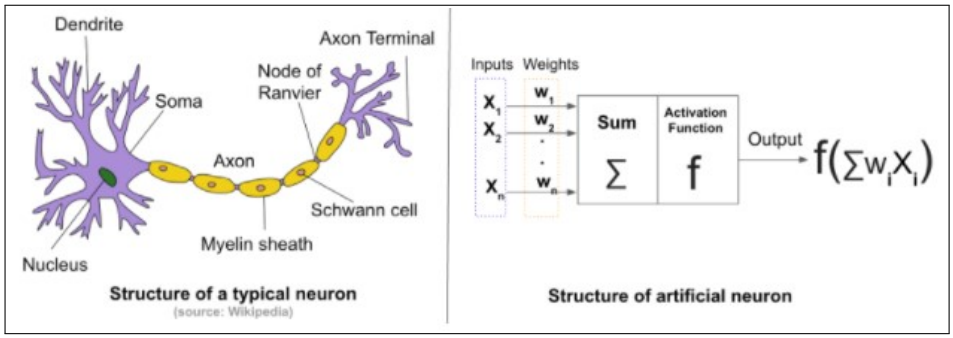


Figure 4.1. Artificial Neuron vs. Biological Neuron

Table 4.1. Artificial Neurons vs. Biological Neurons

Criteria	**Biological Neuron**	**Artificial Neuron**
Typical Operating Speed	$\sim$10 millisecond	$\sim$10 nanosecond
Number of Processing Units	$\sim$10 billion neurons	thousands of processors
Energy Consumption	10^{-16} Joule	10^{-16} Joule Joule
Learning Algorithm	Still unknown	Gradient Descent, back-propagation etc...

Difference between artificial neuron and biological neurons is summarized in Table 4.1.

Working mechanism of artificial neuron is actually very simple. To understand this mechanism, we need to define basic components of artificial neuron: **Input weights** and **biases**, are denoted W_{ij} and b_j respectively. Weights W_{ij} define rate of effect for input x_i from neuron i to neuron j. Bias value b_j allows you to shift the activation function to the left or right, which may be critical for successful learning and it can be considered as preconception in our brain. Another important component of artificial neuron is **activation function** f. Activation function has the role of threshold in biological neuron. Networks are

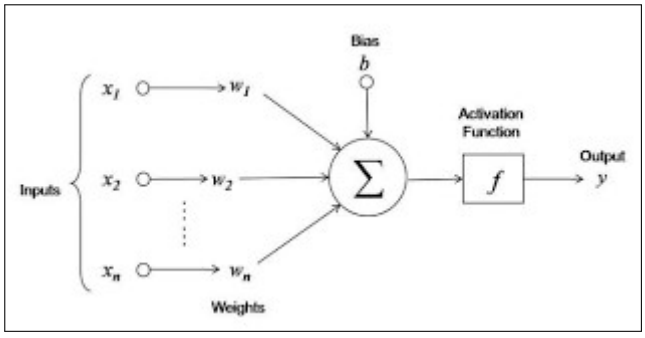


Figure 4.2. Block Diagram of Artificial Neuron

composed of neurons which are stored in **layers** l_i. The first layer that data enter is called input layer. The layer that the prediction or result is given is called output layer. The layer(s) in which actual computation or approximation occurs is called as hidden layer(s). In a layer

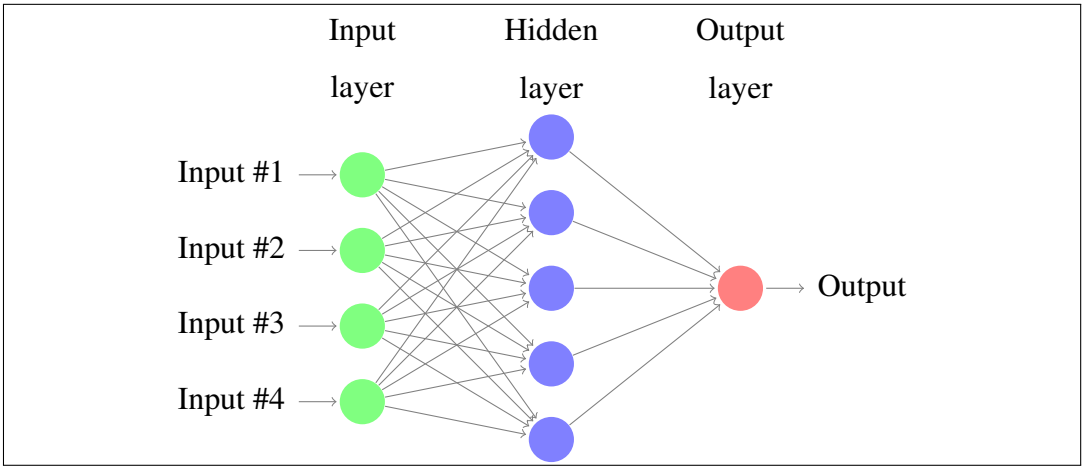


Figure 4.3. Block Diagram of 3 layer neural network

l_i, there can be one or many neurons which are connected neurons from l_{i-1} and l_{i+1} layer. Schematic view of neural network is given in Fig. 4.3. Up to now, the main components of artificial neuron and network are discussed. The most important part of network is the neuron which processes the given input and creates an output.

Suppose n input values x_i are coming to neuron j in l^{th} layer from $(l-1)^{th}$ layer. Then, in a neuron all inputs are processed in the following way:

$$y_{j,t} = b_j + \sum_{i=1}^{n} W_{ij} x_i \tag{4.1}$$

Then, $y_{j,t}$ passes through the activation function f_j and output of j^{th} neuron is denoted by $\hat{o_j}$:

$$\hat{o_j} = f((y_j, t)) \tag{4.2}$$

where W_{ij}, b_j are learned or estimated parameters through the *learning process* in learning cycles or epochs which are the concepts discussed later on. At this step it is important to mention about the role of activation function. Purpose of activation function is to convert the input signal of a node, which is then used as an input in the next layer. Activation function should be applied, otherwise the output will be a linear function. Since neural networks are

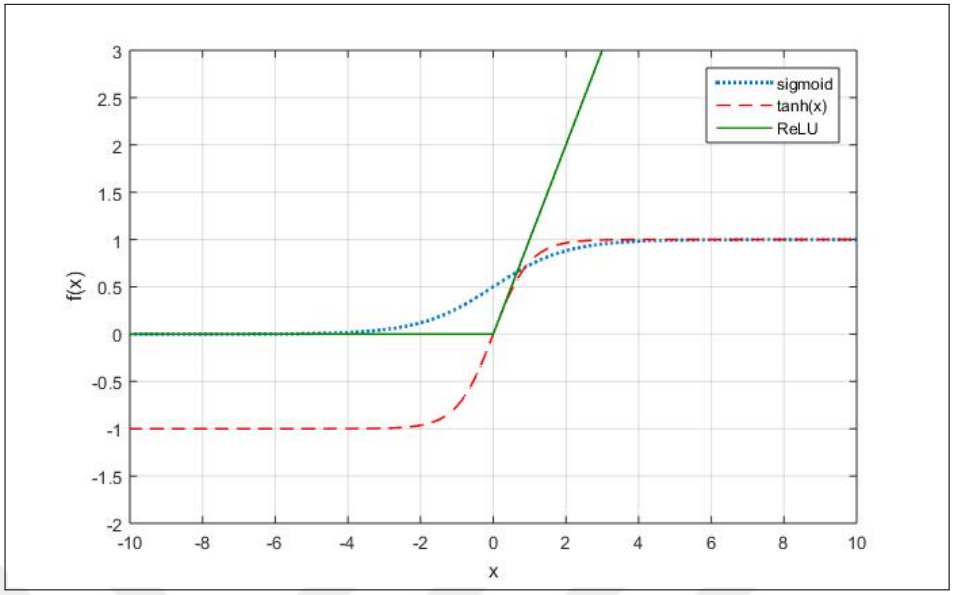


Figure 4.4. Plot of transfer functions: sigmoid, hyperbolic tangent, ReLu

used to work with nonlinear or complicated data sets, such as images, videos, audio, speech, time series, nonlinear activation functions are used. The most commonly used are: sigmoid, hyperbolic tangent, ReLu -Rectified linear units(See Fig. 4.4). Mathematical definition of sigmiod and ReLu functions are given in Eq. 4.3 and Eq. 4.4.The sigmoid is:

$$s(x) = \frac{1}{1 + e^{-x}} \tag{4.3}$$

and ReLu is:

$$r(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{4.4}$$

As you can see from Fig. 4.4, choice of transfer function in neural networks depends on the format of the output from the neuron. If we want artificial neuron which creates outputs as zeros and ones, we will use sigmoid activation function. on the other hand, if we use neural network for forecasting generally, hyperbolic tangent is used in hidden layers and ReLu is used in output layers. Another important feature of artificial neural networks is their ability of learning. In general, there are two general types of learning algorithms: supervised and unsupervised learning. Since we are interested in time series forecasting, we use supervised

learning. Role of weights and biases are already discussed previously and learning in neural network is related to updating connection weights and biases in neuron. Thats why after many training steps, weights and biases are updated according to connection weights and biases in neurons, this causes the network to give true prediction for given input values. But the question is how to update the weights and biases.

The answer is Error (or Cost) Function E:

$$E(y, \hat{o}) = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \hat{o}_i)^2 \tag{4.5}$$

where y is *target variable*. then the weights are adjusted by the following way:

$$W_{ij}^{new} = W_{ij}^{old} - \alpha \frac{\partial E}{\partial W_{ij}} \tag{4.6}$$

where $0 < \alpha < 1$ the learning rate, is generally chosen as 0.01. At this point, it is understood that the activation function should be differentiable. This learning schema is called *back-propagation*. The clever thing about back-propagation is that it enables us to simultaneously compute all the partial derivatives $\frac{\partial E}{\partial W_{ij}}$ in just one pass so the total cost of back-propagation is roughly the same as making just two forward passes through the network.

The learning rate is a relatively small constant that indicates the relative change in weights and biases. If the learning rate is too low, the network will learn very slowly. If the learning rate is too high, the network may oscillate around, overshooting the lowest point with each weight adjustment, but never actually reach it. Some modifications to the back-propagation algorithm allows the learning rate to decrease from a large value during the learning process.

4.1.1. Types of Neural Networks

In the literature, a number of commonly accepted neural network architectures are discussed. In this study, two types of neural networks are used: *Feedforward Neural Network*(FN) and *Recurrent Neural Network*(RN). Feedforward Neural Networks(FNs) are the networks

where connections between neurons in layers do not form a cycle which means the input propagates only in the forward direction (from input layer to output layer). If the network is composed of more than one hidden layer, they are called *multilayer feedforward neural networks* (multilayer perceptrons) denoted by MFN throughout in this study.

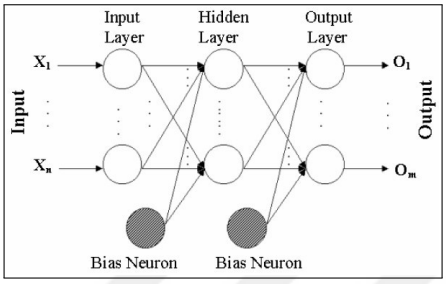


Figure 4.5. Schema of Feedforward Neural Network

When feedforward neural networks are extended to include feedback connections, they are called *recurrent neural networks* (RNs). Since neurons in layer have self-connection, they are considered as networks with a memory. There are some tricky points about neural

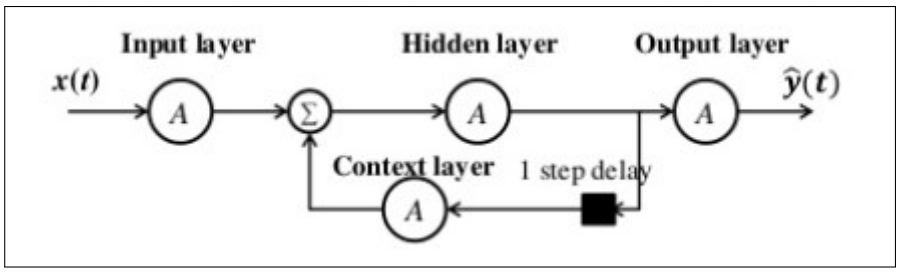


Figure 4.6. Elman's Recurrent Network Diagram

networks. There is no general solution to decide how many hidden layers and how many neurons are needed in networks in deep learning community. But in general, to approximate more complicated dynamics, we need more hidden layers. Although, we mentioned back-propagation learning algorithm, with its variants, so choice of learning algorithm is not unique and depends on the problem on hand.

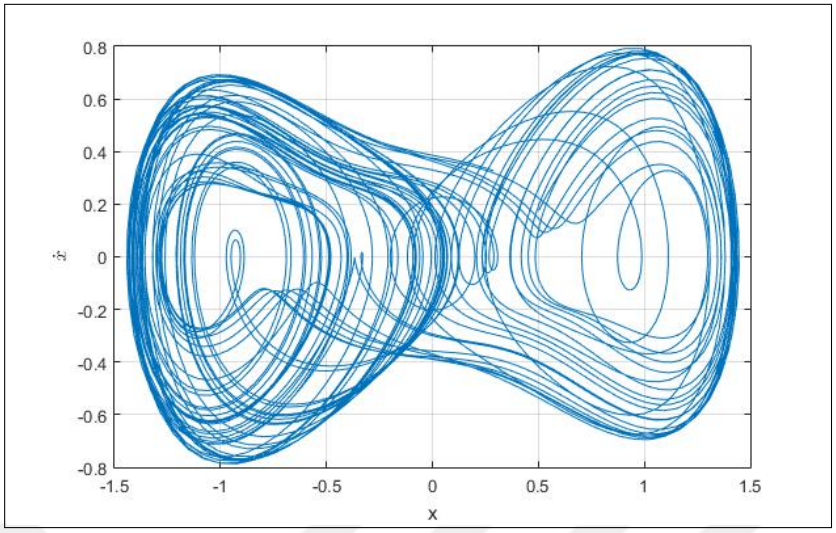


Figure 4.7. Trajectory of Duffing Oscillator in $x - \dot{x}$ plane

4.2. PREDICTION OF CHAOTIC TIME SERIES WITH NEURAL NETWORKS

In order to compare the capability of neural networks in forecasting chaotic time series, three different dynamical systems are used: Duffing oscillator, Rössler System and MYMH system which we analyzed in Chapter 3. In the following two subsections, governing equations of each system are given and with three types of neural networks, forecasting performance of each network architecture is compared. This makes use of numerical solutions yielded by systems.

4.2.1. Duffing Oscillator

The chaotic dynamics of Duffing oscillator has been studied in many works in literature. The Duffing equation that is used in this study is:

$$\ddot{x} + \delta\dot{x} + \beta x + \alpha x^3 = \gamma sin(wt) \tag{4.7}$$

The given system (Eq. 4.7 displays chaotic behavior with for parameter values $\gamma = 0.42$, $\delta = 0.5$, $\alpha = -1$, $\beta = 1$,$w = 1$ and initial conditions $(x_0, \dot{x}_0) = (0.5021, 0.17606)$ and trajectory of the system is given in Fig. 4.7.

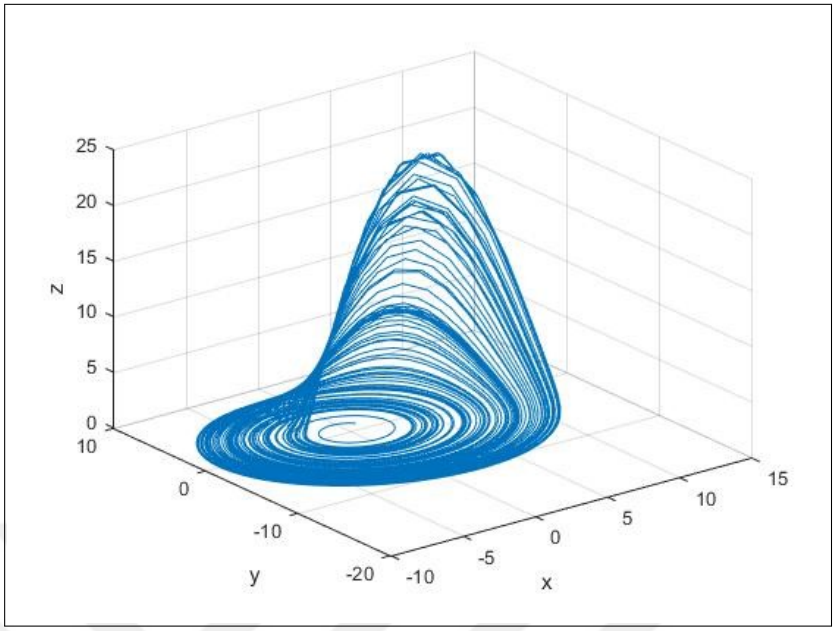


Figure 4.8. Rössler attractor for given set of initial conditions and parameters

4.2.2. Rössler System

The second system we are interested in is the Rössler system or attractor:

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c) \end{aligned} \tag{4.8}$$

Where a, b, c are real constant parameters of the system. For $a = b = 0.2$, $c = 5.7$, the system has corresponding Lyapunov exponents: $(0.0714, 0, -5.3943)$. Since there exists a positive Lyapunov exponent the system displays chaotic behavior. (see Fig. 4.8)

4.2.3. Use of Neural Network Estimating Chaotic Time Series Generated From Dynamical Systems

The systems that we are going to use for comparing the capability of neural networks in forecasting chaotic time series were defined in subsections 4.2.1 and 4.2.2. In this subsection, we compare the performance of three different neural networks: multilayer feedforward

neural network (MFN), single layer feedforward (FN) and single layer recurrent neural network (RN).

In this study, we have used firstly, feedforward neural network with one hidden layer with m number of input units. Duffing oscillator is two-dimensional system $(d = 2)$. According to Takens theorem, an embedding dimension must be less than or equal to $2d + 1$. We choose embedding dimension $m = 4$. Then we use multilayer feedforward neural with the following architecture $(m : 2m : m : 1)$ which is the same architecture used in [37]. As a third architecture we use Elmans recurrent neural network with one hidden layer.

In our simulations, performances of the three mentioned neural networks are compared using data with and without additive noise. For performance comparison of neural network without noise case we let $y_t = x_t$ where y_t is our target variable. For the noisy data generation, we add noise to our data in the following way:

$$y_t = x_t + u_t \tag{4.9}$$

Where u_t represents the noise component and $u_t = \eta\sigma g_t$ where σ is sample standard deviation of x_t and η is noise level which takes different values between $0 < \eta < 1$ and g_t is standard normal random variable.

In the result, for forecasting performance of each network root-mean-square error (*rmse*) is used as performance criteria. Using the multilayer feedforward neural network (MFN) with architecture $(4 : 8 : 4 : 1)$, the neural network predicts the actual time series perfectly with *rmse* $= 3.003 \times 10^{-5}$ for Duffing oscillator (without noise) and predicted and original time series are plotted in Fig. 4.9. For Rössler system, we use the MFN with architecture $(4 : 8 : 4 : 1)$ and for without noise, the network is very efficient. In Fig. 4.10 , predicted and original time series are for Rössler system without noise case is given. For the same data set, generated from Duffing system, we use Elmans recurrent neural net in predicting Duffing oscillator, which is very similar to single hidden layer feedforward neural network, but it also uses previous estimation value as an input which makes it a neural network with a memory. For single hidden layer we try to choose optimal number of neurons. We choose

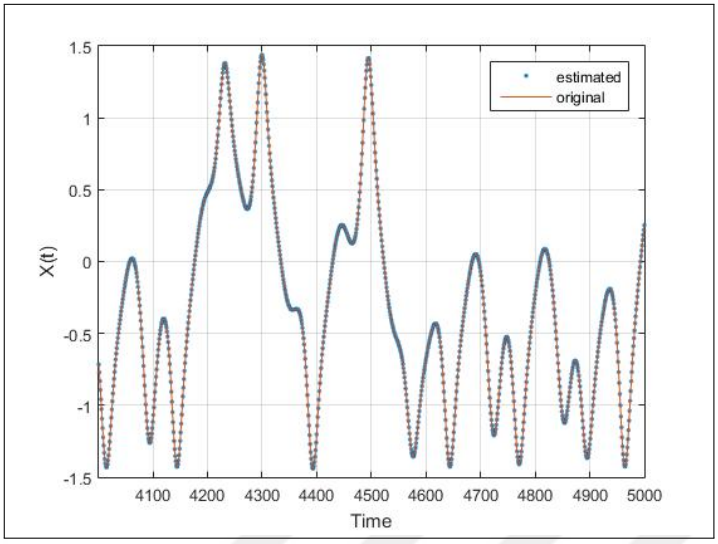


Figure 4.9. Prediction of Duffing time series without noise for $MFN(4:8:4:1)$

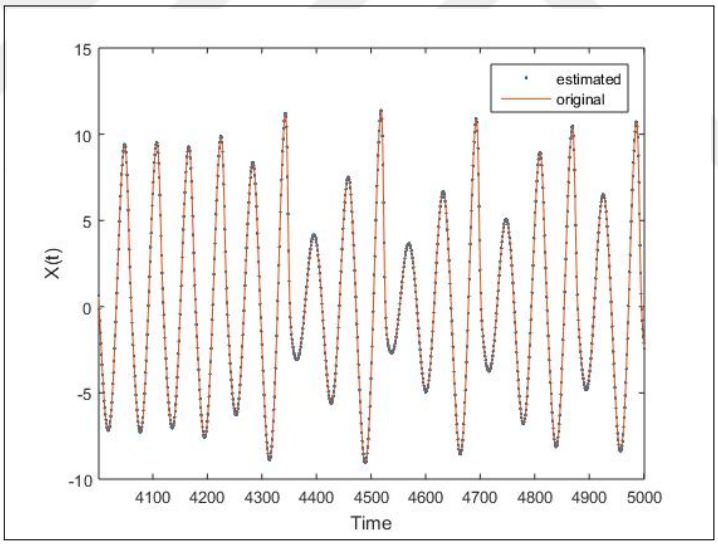


Figure 4.10. Prediction of Rössler time series without noise for $MFN(6:12:6:1)$

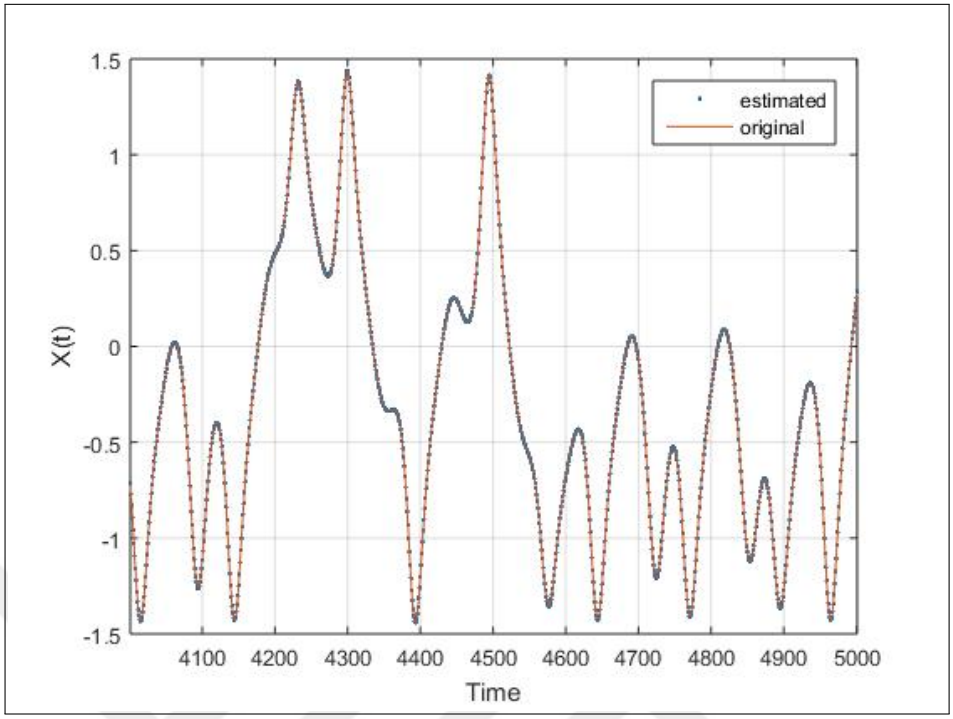


Figure 4.11. Recurrent neural network estimation of Duffing system where $\eta = 0$

optimum number of neurons as 8 which is two times the embedding dimension $m = 4$. For forecasting result, Elman network display performance as good as multilayer feedforward neural network. Elmans neural network gives *rmse* $= 3.349 \times 10^{-5}$. Estimation of Elmans neural network and original data is plotted in Fig. 4.11. It is also important to note that with the given architectures of networks, Elman network gives results for smaller number of epochs than the multilayer feedforward one.

Finally, for the Duffing oscillator without noise, we test the performance of feedforward neural network with single hidden layer. For the single hidden layer, number of neurons is chosen as again $2m$ which is same as Elman recurrent network. With single hidden layer neural network, performance of forecast is slightly increasing compared to recurrent one with *rmse* $= 3.328 \times 10^{-5}$ but it is worse than the multilayer neural network. In addition, the speed of convergence of single hidden layer feedforward neural network is higher than the recurrent one. For Rössler system, recurrent network with 8 neurons in hidden layer gives the optimum performance. Plot of estimation of $RN(12)$ given in Fig. 4.13 (with *rmse* $= 3.399 \times 10^{-5}$).

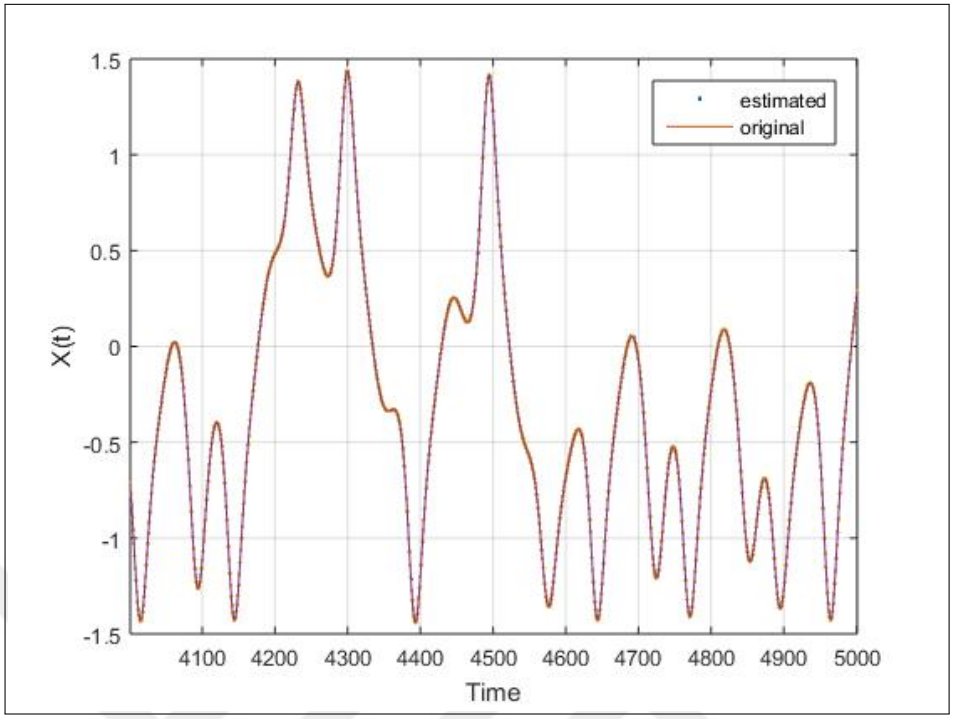


Figure 4.12. Estimation of $FN(12)$ network topology for Duffing system where $\eta = 0$

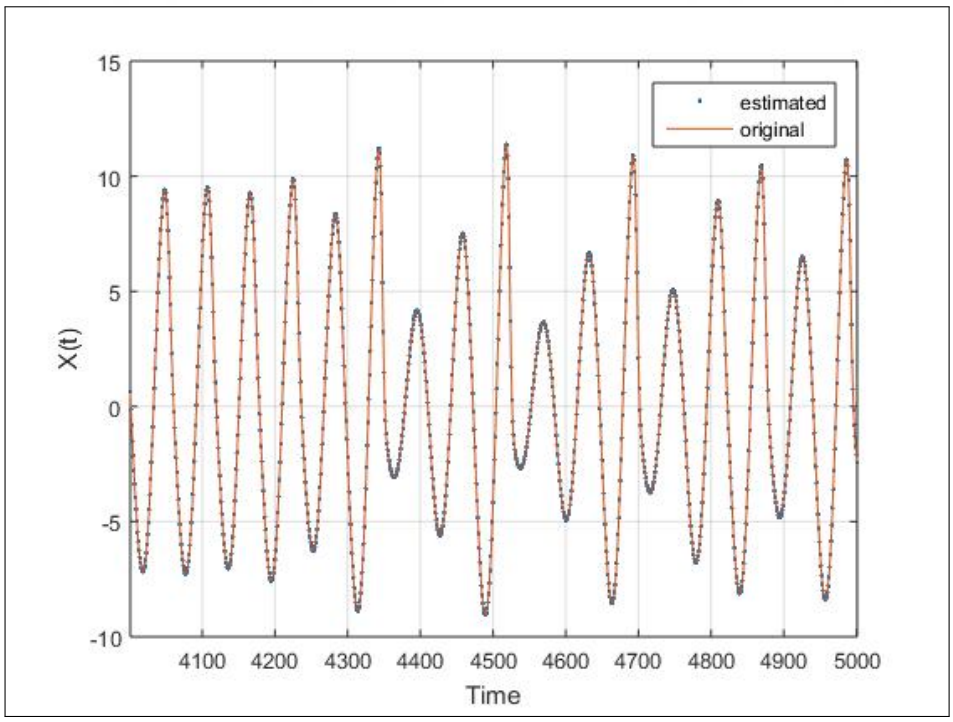


Figure 4.13. Estimation of $FN(8)$ network topology for Rössler system where $\eta = 0$

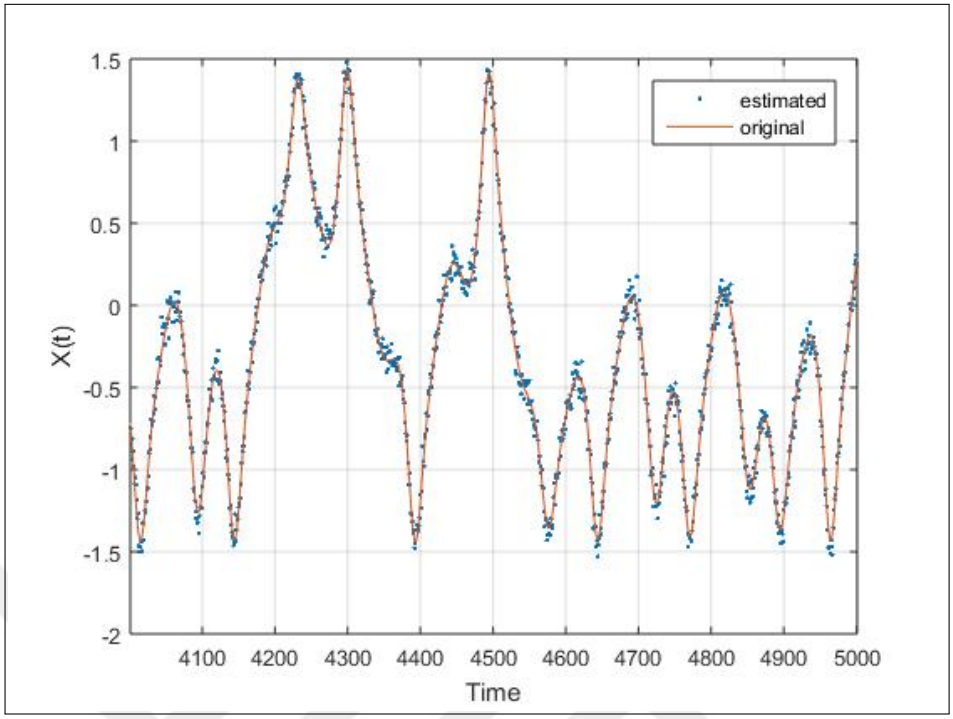


Figure 4.14. Duffing system prediction of $MFN(4:8:4:1)$ with $\eta = 0.01$

After forecasting the Duffing data set without noise, we test the noise filtering capabilities of these three types of neural networks. For $\eta = 0.01$, multilayer neural network, forecast performance is good but as it is expected it is worse than the case without noise (with *rmse* $= 0.059$). Estimated and original data is plotted in Fig. 4.14. For $\eta = 0.1$, performance of multilayer feedforward decreases as expected compared to previous case (Fig. 4.15). It gives *rmse* $= 0.5755$. For $\eta = 0.2$, *rmse* increases to 0.765(see Fig. 4.16).

To increase the forecasting performance of MFN, we increase neuron numbers in layer from 4 to 5. With this change, we train the network and make prediction for data with error rate $\eta = 0.2$. Neural network gives better estimate with *rmse*=0.6635 and forecast of modified MFN is plotted in Fig. 4.17. After analyzing noise filtering performance of MFN, we apply the same tests to Elman type recurrent network. In [22], they demonstrated that recurrent neural networks are more accurate than single layer feedforward neural networks. In our analysis, we observe that for $\eta = 0.01$, it performs a little bit better than MFN but there is no significant improvement (with *rmse*=0.053). When we increase the error rate to $\eta = 0.1$, performance of recurrent network is again close to MFN (with *rmse*=0.5498). For $\eta = 0.2$,

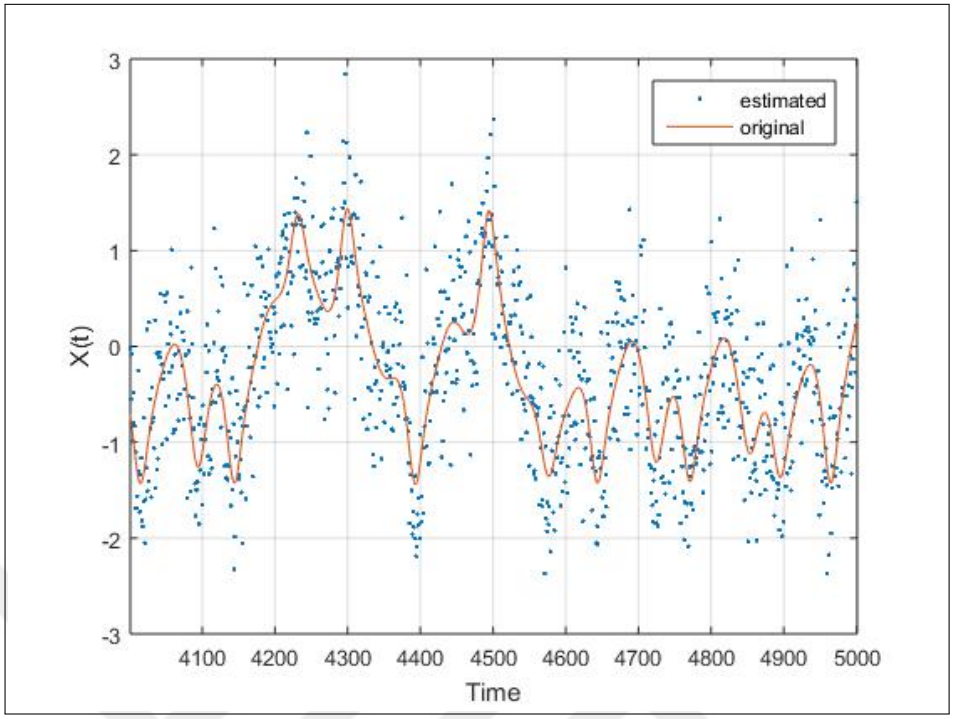


Figure 4.15. Duffing system prediction of $MFN(4:8:4:1)$ with $\eta = 0.1$

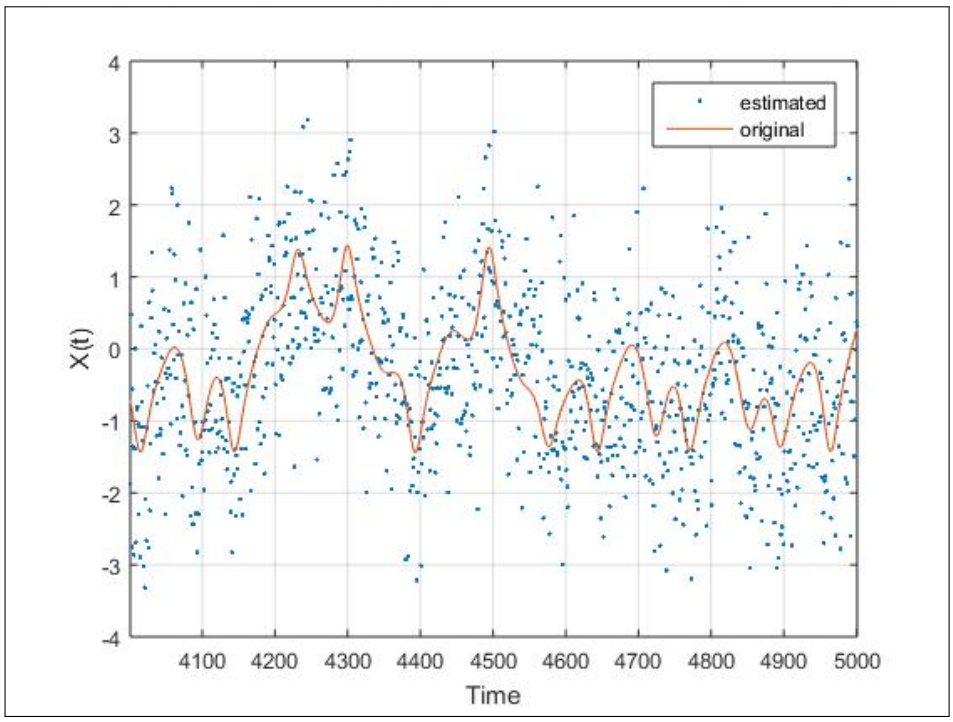


Figure 4.16. Duffing system prediction of $MFN(4:8:4:1)$ with $\eta = 0.2$

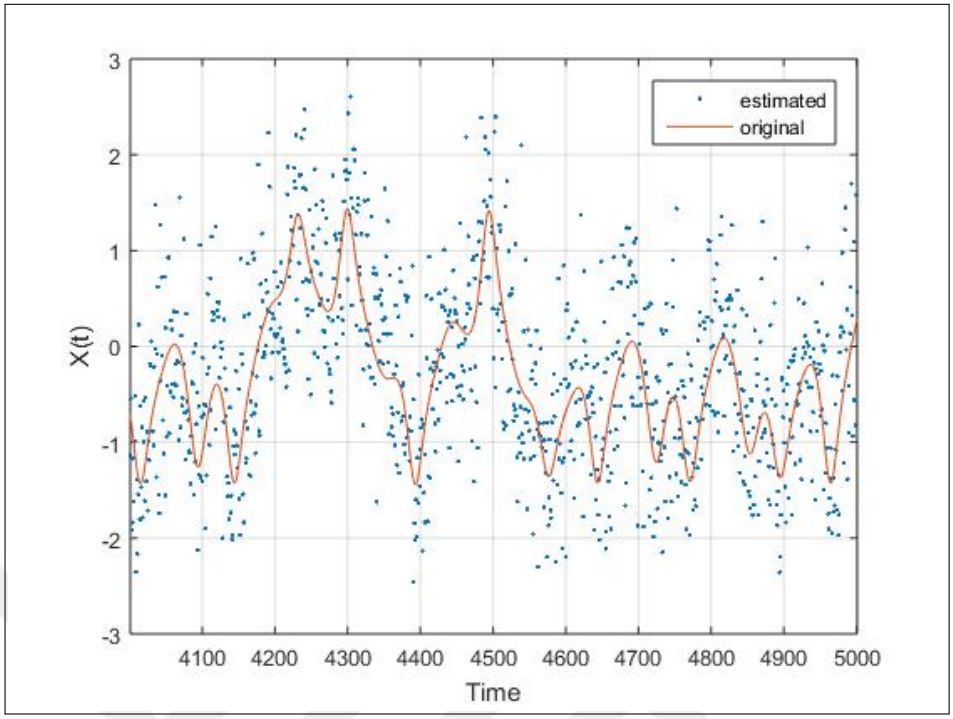


Figure 4.17. Duffing system prediction of $MFN(5:10:5:1)$ with $\eta = 0.2$

performance of recurrent network clearly better than MFN with *rmse*=0.6487. Prediction of recurrent network with error rate $\eta = 0.2$, is plotted in Fig. 4.18.

$FN(8)$, for time series with error rate $\eta = 0.01$, performs as well as both of the other two neural nets. For $\eta = 0.1$, performance of this network is worse than both recurrent and MFN' one. Finally, for $\eta = 0.2$, forecast performance is getting worse than previous cases as it is expected.

It can be seen in previous analysis results, as mentioned in [22], RN is better predictor model than FN in noisy time series. For higher noisy level ($\eta = 0.2$) interestingly performance of MFN is not as good as RN and FN. In Table 4.2, *rmse* values of each prediction for Duffing system is given.

For the Rössler system, we test each network architecture. In summary, Rössler system can also be estimated very efficiently by the described network architecture. Like Duffing oscillator, when we increase the noise level, prediction error rates increase. In Table 4.3,

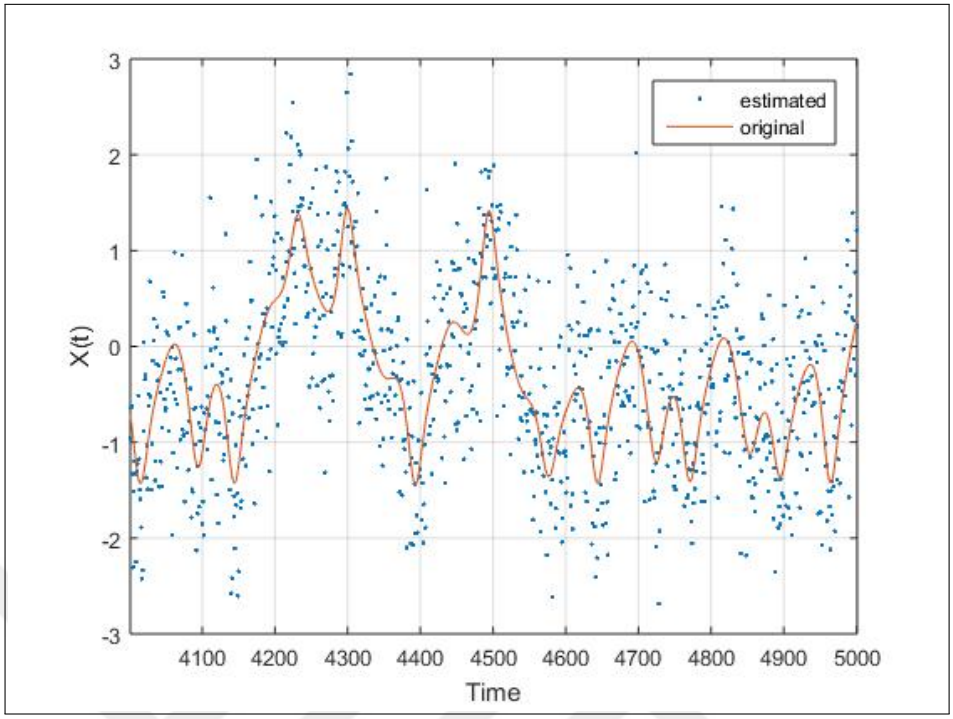


Figure 4.18. Duffing system prediction of $RN(8)$ with $\eta = 0.2$

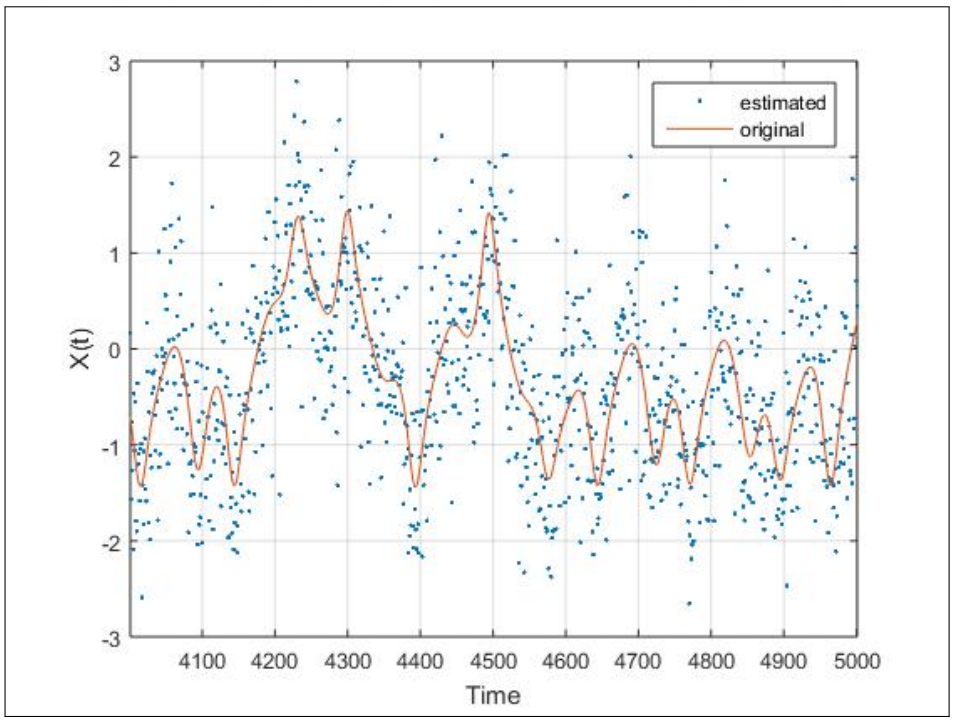


Figure 4.19. Duffing system prediction of $FN(8)$ with $\eta = 0.2$

Table 4.2. *rmse* values of each network with respect to changing error rate η for Duffing oscillator

Network Model	***rmse***
$\eta = 0$ (no noise)	
MFN(4:8:4:1)	3.003×10^{-5}
RN(8)	3.349×10^{-5}
FN(8)	3.328×10^{-5}
$\eta = 0.01$	
MFN(4:8:4:1)	0.059
RN(8)	0.053
FN(8)	0.058
$\eta = 0.1$	
MFN(4:8:4:1)	0.5755
RN(8)	0.5498
FN(8)	0.5997
$\eta = 0.2$	
MFN(4:8:4:1)	0.7650
RN(8)	0.6487
FN(8)	0.6766

rmse error values for each prediction of Rössler system are given.

4.3. ALGORITHM FOR CALCULATION OF LYAPUNOV EXPONENT USING ARTIFICIAL NEURAL NETWORK

In general, since people have rarely known the real generating function G of a given time series, there is an alternative point of view: Suppose there is an observation function $h : \Re \to \Re$ that generates our observation y_t:

$$y_t = h(x_t) \tag{4.10}$$

Table 4.3. *rmse* values of each network with respect to changing error rate η for Rössler System

Network Model	***rmse***
$\eta = 0$ (no noise)	
MFN(6:12:6:1)	3.3625×10^{-5}
RN(12)	3.399×10^{-5}
FN(12)	3.388×10^{-5}
$\eta = 0.01$	
MFN(6:12:6:1)	0.048
RN(8)	0.053
FN(8)	0.054
$\eta = 0.1$	
MFN(4:8:4:1)	0.8755
RN(8)	0.7498
FN(8)	0.8997
$\eta = 0.2$	
MFN(4:8:4:1)	0.9650
RN(8)	0.8487
FN(8)	0.9887

And all that is available for us is the sequence y_t .With time lag or embedding dimension m, the new sequence is generated as:

$$y_t^m = (y_t, y_{t+1}, \ldots, y_{t+m-1}) \tag{4.11}$$

Using these identities, we get the following result:

$$J^m(G(x_t)) = g(J^m(x_t)) \tag{4.12}$$

So g is topological conjugate of G. That's why, g has the same dynamical properties of G.

So the mapping g can be estimated as:

$$g : \begin{bmatrix} y_t \\ \vdots \\ y_{t+m-1} \end{bmatrix} \rightarrow \begin{bmatrix} y_{t+1} \\ \vdots \\ \nu(y_t, y_{t+1}, \ldots, y_{t+m-1}) \end{bmatrix} \tag{4.13}$$

This result reduces to estimating:

$$y_{t+m} = \nu(y_t, y_{t+1}, \ldots, y_{t+m-1}) \tag{4.14}$$

At this step, ν is estimated by neural network. Linearization of map G gives:

$$\Delta y_{t+1}^m = (DG)_{y_0^m} \Delta y_0^m \tag{4.15}$$

Then the solution is:

$$(Dg^t)_{y_0^m} = (Dg) y_{t-1}^m ... (Dg) y_0^m \tag{4.16}$$

Table 4.4. Lyapunov exponents of systems calculated by both TISEAN and FN networks

	Lyapunov Exponents
	Rössler System
TISEAN results	(0.0714, 0, -5.3943)
FN(12)	(0.0704, 0.002, -5.4943)
	Duffing System
TISEAN result	(0.3675, 0.4256)
FN(8)	(0.3254, 0.4613)
	MYMH System
TISEAN results	(0.0011, 0.00035,-0.00035,-0.0011)
FN(16)	(0.0015, 0.00041,-0.00043,-0.0007)

and

$$(Dg)y_t^m = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & 0 & 1 \\ \nu_{1,t} & \nu_{2,t} & \nu_{3,t} & \dots & & \nu_{m,t} \end{bmatrix} \tag{4.17}$$

Then the Lyapunov exponents can be calculated from the eigenvalues of matrix in Eq. 4.17. To test the efficiency of the given algorithm, we test it by using the time series generated from Rössler (Eq. 4.8), Duffing (Eq. 4.7) and MYMH(Eq. 2.17) systems. As we already stated that ν in Eq. 4.14 is estimated by the neural network. For calculating maximal Lyapunov exponents of the systems, we use single hidden layer feedforward neural network. Estimation of maximal Lyapunov exponents by neural network and Lyapunov exponents values calculated using TISEAN package[38](using $lyap_k$ function) is compared to reach a conclusion about the validity of results obtained by neural networks. In Table 4.4, Lyapunov exponents of three system are displayed for the test systems. In the estimation of Rössler system 12 hidden units and 2000 points generated from Eq 4.8 are used in a single hidden layer feedforward network. We find the Lyapunov exponents of the Rössler system using

$FN(12)$, this gives Lyapunov exponents of the system as $(0.0704, 0.002, -5.4943)$ which is very good estimate without noise condition ($\eta = 0$). For the Duffing system, best estimates are observed with 8 hidden units for 2000 observations with feedforward network. For this analysis, estimates are not as accurate as such in the Rössler case. Yet, they are not too much different from the expected values. For MYMH system, we use the initial conditions(p_x, p_y, x, y)=(0.2440286868, 0.2, 0.0, -0.1) and parameters(g=0.25, a=1). For calculation of Lyapunov exponents,we use again 2000 points and with network of 16 units.As we already mentioned MYMH system displays both chaotic and regular behaviors at the same time.

4.4. MODELING DYNAMICAL SYSTEM WITH NEURAL NETWORK

In this chapter, up to now, we use neural networks for modeling nonlinear time series by using previous time steps. Neural networks can also be used to predict a value of a target variable which is the result of other parameters. For example, suppose you want to estimate how many kilograms of rain per meter-square fall in İstanbul during August. There exist historical rainfall values in hand, data of the factors that affect the rainfall amount such as humidity, wind etc... However, writing a mathematical model which describes the rainfall amount is not an easy task. As we have already mentioned neural networks are able to estimate very complicated functions. In this section, we test the performance of neural networks in modeling dynamical systems.

We use the Rössler (Eq. 4.8) and Duffing system (Eq. 4.7) as test cases for neural networks.During the analysis, the time series generated in section 4.2 for both system are used. For modeling systems, many types of neural networks are studied: RBF neural networks[39], nonlinear auto regressive model with external input (NARX) neural networks[40]. In this study, we compare the performance two neural networks on modeling dynamical system: FN and $NARX$ neural networks. In section 4.1.1 definition of FN is given. However, we did not yet define $NARX$.

The NARX is a type of RN. This network model creates relationships between current and past values of the driving (exogenous) series and past values of the target time series which

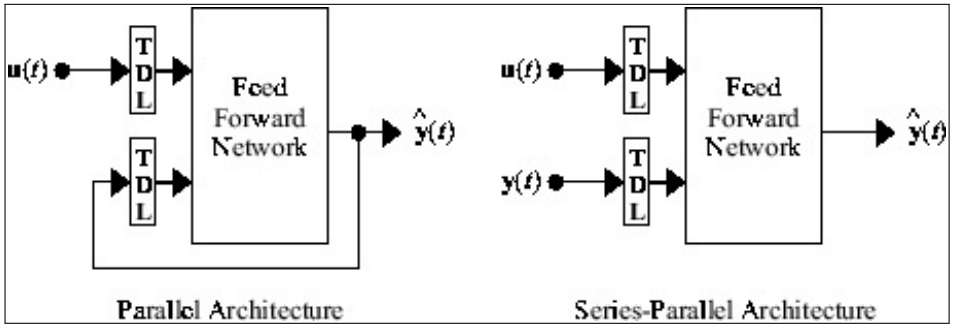


Figure 4.20. Types of NARX networks

can be represented mathematically by the following notation:

$$y_t = F(y_{t_1}, y_{t-2}, y_{t-3}, \dots, u_t, u_{t-1}, u_{t-2}, u_{t-3}, \dots) + e_t \tag{4.18}$$

where y_t is again target variables and u_{t-i} and e_t denote values of driving variables and error. NARX network has two types of architecture, which are shown in Fig. 4.20. The difference between these two architectures can summarized as follows: while the target time series y_t in series-parallel architecture are estimated using present and past values of u_t and real past values of y_{t-1}, in parallel architecture,the predicted past values of y_{t-1} are used instead of real past values y_{t-1} along with u_t. Inner learning mechanism of NARX networks are same as other neural networks such as in FN and RN.

In the analysis, we used Levenberg-Marquardt learning algorithm for both system in neural networks. For Rössler system, we use 6 hidden neurons in both FN and NARX networks. In training state of networks 70% of data is used and 15% of data is used for as test case, the remaining 15% of data is used as validation set. As performance criteria,we use *rmse* of estimates. The test results for both FN and NARX networks are plotted in Fig. 4.21 and it is clear that the variables $x(t)$ of Rössler system can be estimated very well by the both network architecture. While *rmse* of NARX network is 3.22×10^{-4} ,for FN, *rmse* is equal to 4.07×10^{-4}. In addition to this, other variables of the system, namely $y(t)$ and $z(t)$, are studied using the same network structures and the success of both network are also observed in these scenarios.

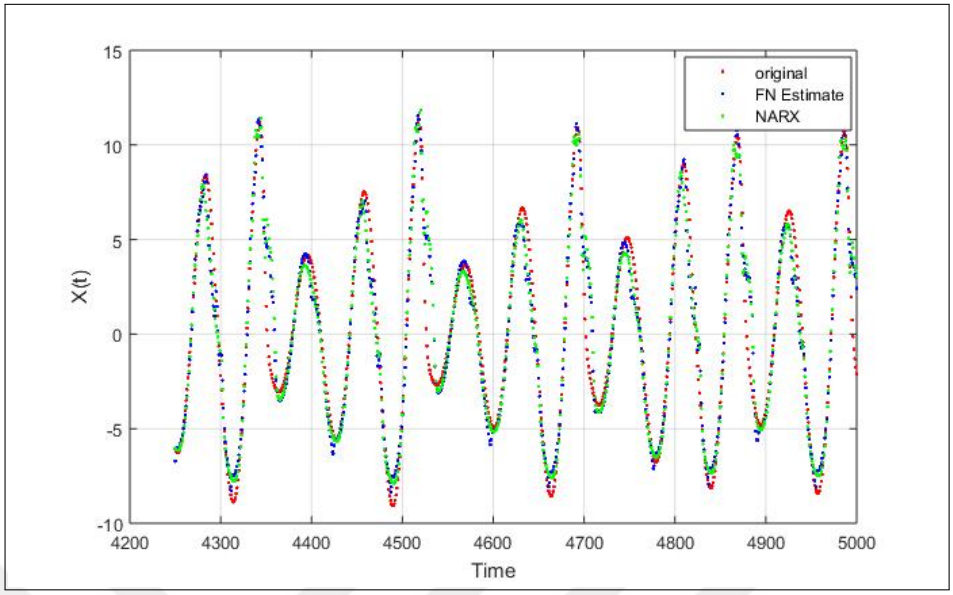


Figure 4.21. Predicted x(t) values of FN network vs. predicted x(t) values NARX Network for Rössler System

For Duffing system, we use single hidden layer NARX network with 4 neurons, again we use tangent-sigmoid activation function in hidden layer and linear activation function in output layer. Output of $NARX$ and FN networks for Duffing system are plotted in Fig. 4.22. For Duffing system, we want to try modelling $\dot{x(t)}$ in Eq. 4.7 based on the variable $x(t)$ for Duffing system, note that time dependent part of $\ddot{x}$ is not considered in neural network model input but both networks are very successful in modeling system variables based on the other dependent variable. For $NARX$ network, the *rmse* is equal to 2.22×10^{-5} and for FN network *rmse* is 2.82×10^{-5}.

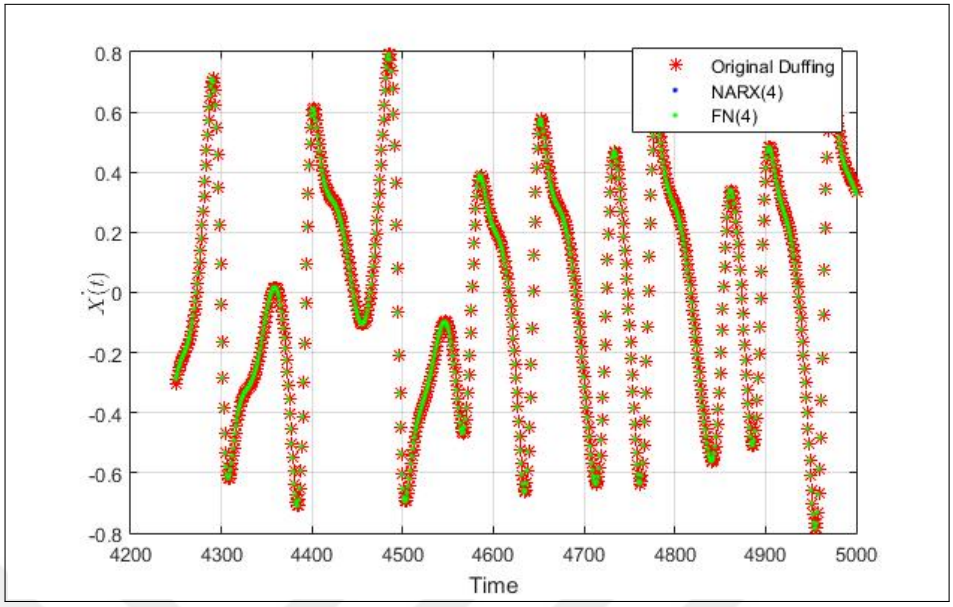


Figure 4.22. Predicted $\dot{x(t)}$ values of FN network vs. predicted $\dot{x(t)}$ values NARX Network for Duffing System

5. CONTINUOUS DYNAMICAL SYSTEMS AS PSEUDO RANDOM NUMBER GENERATOR

Random numbers are used in many fields of study ranging from generating data encryption keys to simulating and modeling complex phenomena. There are two main categories of random numbers: those generated by Pseudo Random Generators (PRNGs) and True Random Number Generators (TRNGs). TRNGs (or hardware based RNGs) are based on physical processes such as radioactive decay, photoelectric effect, thermal noise, atmospheric noise etc. TRNGs have the quality required for random numbers: they are nondeterministic and unpredictable. On the other hand, they have two main drawbacks: they are slow, and they require good hardware implementation which is costly.

The second alternative to create random numbers is PRNGs. PRNGs uses mathematical formulae to create random sequences. PRNGs are very efficient which means they can generate many random numbers in short time but if one seeks an undeterministic way of generation they are not suitable since they are created by a deterministic approach. That's why, they would be predictable and repeatable or periodic. In the light of this information, desired PRNGs should satisfy the following criteria: they must create uniformly distributed, fast and serially uncorrelated random numbers with long periods. There are many algorithms used as PRNGs to generate numbers that appear as random; the well-known example of PRNGs is linear congruential generator (LCG). LCGs are fast but they are not a good choice for Monte-Carlo simulations and cryptographic applications. LGC does not satisfy the high criteria of randomness because of their short period and serial correlation between successive random numbers.

Chaotic systems are recently used as a random number generator especially in cryptography and data encryption. Researchers prefer to use chaotic dynamical systems as PRNG due to their non-periodic behavior and they can be used as fast random number generators. Chaos theory aims to explain the long-term behavior of a system in a deterministic way. One important characteristic of the chaotic systems is their sensitive dependence on its initial conditions which means if one wants to integrate equations of motion of a chaotic system,

even an infinitesimal change in the initial conditions will increase the deviation exponentially over time and will easily collapse the accuracy of the prediction. That's why chaotic systems are unpredictable which makes them attractive for pseudo random number generators.

Iterative maps are discrete continuous systems which are used as PRNGs in many studies. Logistic map is studied as a number generator in [41], 3-d chaotic Arnold cat-map is studied in [42]. The chaotic Henon map was used as a RNG in [43]. The most famous example of chaotic systems is the Lorenz System. Lorenz system was used as a random number generator in [44, 45] and in [45],it is shown that Lorenz system could be a good candidate for a desirable random generator. The Rössler system as PRNG is described in [46]. Chaotic Chua's circuit is also another example of dynamical system as PRNG studied in [47].

To test the PSRGs, there are some statistical test suites such FIPS-140-2, DIEHARD and NIST test suites. In in this chapter, we propose generator (PRBG) based on two chaotic continues dynamical systems namely: Rössler System(Eq. 4.8) and Duffing oscillator(Eq. 4.7).

5.1. DESCRIPTIONS OF STATISTICAL TESTS FOR RANDOMNESS

There are many statistical tests for quality of random numbers: Poker test, Runs test, Frequency Test etc. In this analysis we use a NIST test suite for statistical analysis of our new PRNG described in section 5.2. For the usage of our PRNG in image encryption,we use other statistical test. In this section, we briefly introduce the test that we use in this study.

5.1.1. Histogram Analysis

As we have already know images are consisting of pixels. Each pixel,if the image is black and white, has its own grey scale level. So if we plot the number of pixels in each grey scale level, we will get the histogram of an image. The encrypted images are called as *cipher-image*. The good image encryption algorithm creates a cipher-image which has uniform distribution in histogram. In addition to this, the histograms of cipher image and original image should be significantly different. Histogram analysis gives information about

confusion and diffusion properties of cipher-image and it is important to characterize the power of proposed solution against statistical attacks.

5.1.2. Information Entropy Analysis

Concept of entropy is very familiar and important measure for physicists from thermodynamics. Entropy is measure of degree of disorder in a system. In information theory, Shannon[48] defines the entropy as the average minimum amount of information needed to encode a given data sets(string and image), upon the frequency of the given system. Entropy gives information about the degree of unpredictability. That's why, the higher the entropy, the better the security. Entropy $E(X)$ for source x defined as:

$$E_(X) = \sum_{i=1}^{n} P(x_i) \log_2 \left(\frac{1}{P(x_i)} \right) \tag{5.1}$$

where X refers to our image(pixel value matrix), x_i denotes random pixel in X and $P(x_i)$ is the probability of choosing x_i in X. For a source emitting 2^N symbols, the maximum value of entropy can be N.

5.1.3. Correlation of Adjacent pixels

There is a high correlation between adjacent pixels in each direction(horizontal, vertical, diagonal) for an ordinary image. Yet, an encrypted image should have low correlation between adjacent pixels in each direction. Correlation between adjacent pixels can be tested visually by plotting distribution of adjacent pixels of original image versus cipher image. Numerically measure of correlation coefficient between adjacent pixels $r_{x,y}$ can be calculated using:

$$r_{x,y} = \frac{cov(x,y)}{\sqrt{V(x)}\sqrt{V(y)}} \tag{5.2}$$

where $cov(x, y)$ denotes co-variance between x and y values of pixels:

$$cov(x, y) = M(x - M(x))(y - M(y)) \tag{5.3}$$

$M(x)$ is the average of X_i and $V(x)$ is the variance of X_i.

$$M(x) = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{5.4}$$

$$V(x) = \frac{1}{N}\sum_{i=1}^{N}(X - M(X_i))^2 \tag{5.5}$$

5.2. PROPOSED PRNG

Our proposed random number generator is a kind of LCGs, based on integration of equations of the system in time. We choose a dynamical system:

$$\dot{x} = f(x) \tag{5.6}$$

$x \in R^n$. Then the system is integrated in time. As a second step, we define functions $S(t_i)$ and $M(t_i)$ as follows:

$$S(t_i) = \sum_{j=1}^{n} 10^{14} x_j(t_i) \tag{5.7}$$

$$M(t_i) = floor(S(t_i))\ mod\ 2^d \tag{5.8}$$

where $x_j(t_i)$ is j^{th} component of the system Eq. 5.6 at i^{th} iteration in time t. We choose 10^{14} as multiplier to make period of our PRNG long. In Eq. 5.8, $M(t_i)$ takes values between

$(0, 2d-1)$. At the final step $M(t_i)$ is converted to d-bit binary sequences. In our test we choose $d = 8$ and generated random integers from time series between $(1, 255)$.

5.3. STATISTICAL TESTING OF PROPOSED PRNGS

To be sure that the newly proposed pseudo random number generators are cryptographically secure, they should be subjected to a variety of statistical tests designed to detect the specific characteristics expected of truly random sequences. The four most generally used test suites are: NIST [49], DIEHARD [50], Crypt-XS, FIPS-140-2. In this study, we use NIST test suite. The NIST tests suite is a statistical package which consists of 16 tests that are developed to test the randomness of (arbitrary long) binary sequences produced by either hardware or software based cryptographic random or pseudo random number generators.

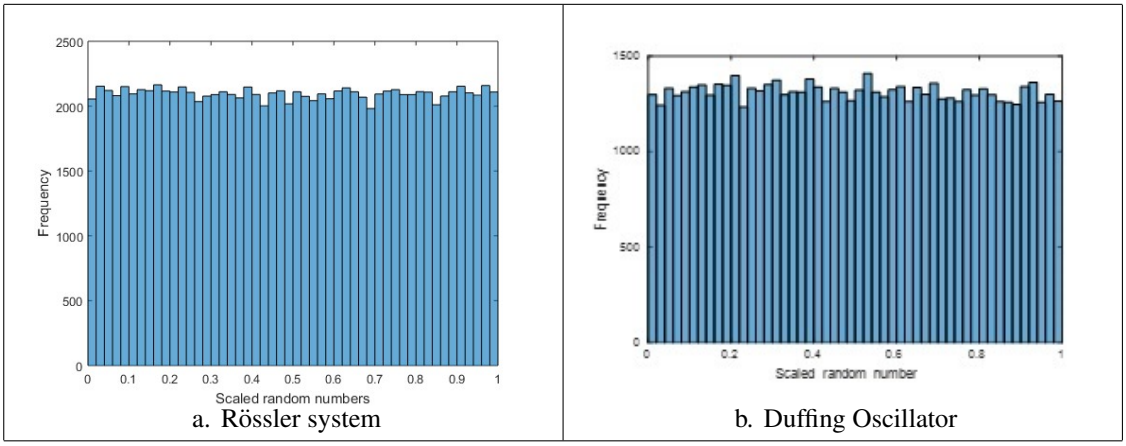
For generation of time series, we integrate given two systems in time and we generate the random numbers by the proposed method in section 5.2. For the Rössler system (Eq. 4.8), we use the following initial conditions and parameter values: $(x_0, y_0) = (0.5021, 0.17606)$ and a = 0.2, b = 0.2, c = 5.7. For the Duffing oscillator, the initial condition is $(x_0, y_0) = (0.5021, 0.17606)$ with parameter values γ=0.42, δ=0.5, α=-1, β=1, w=1. Both systems are integrated in time and we have time series with length 2000 for both of the system. Then these random number are converted to 8-bit sequences. Finally, these data sets are tested using NIST test suite. Test results with p-values for both systems is given in Table 5.1. In addition to this, distribution of generated random numbers with two systems mentioned are shown in the histograms in scaled range $(0, 1)$ in Fig. 5.1. According to these histograms both generated PRNGs obey uniform distribution.

5.4. USE OF CPRNGS IN IMAGE APPLICATION

To test our chaotic PRNGs given above, we use the algorithm schema used in[51] for encrypting the images by adapting our CPRNGs and evaluate their performance in terms of their ability of the attackers to break it by using the following three methods: histogram analysis, entropy analysis and correlation analysis. The image encryption algorithm that we use is based on perturbation of image by shuffling (by XOR operator) of row, columns and

Table 5.1. Results of the SP 800-22 test of PRNG for Rössler and Duffing oscilattor based on PRNGs

	Rössler System		**Duffing Oscillator**	
Test index	***p-value***	***Result***	***p-value***	***Result***
Frequency	0.5436	Success	0.5436	Success
Block frequency	0.6784	Success	0.4217	Success
Cumulative sums	0.4327	Success	0.2561	Success
Runs	0.6921	Success	0.4378	Success
Longest run	0.1121	Success	0.0914	Success
Rank	0.7221	Success	0.5412	Success
Non-overlapping template	0.0322	Success	0.0355	Success
FFT	0.6121	Success	0.3421	Success
Overlapping template matching	0.1398	Success	0.1021	Success
Universal	0.2655	Success	0.1465	Success
Approximate entropy	0.2356	Success	0.1137	Success
Random excursions	0.3102	Success	0.2102	Success
Random excursion variant	0.0543	Success	0.0431	Success
Serial	0.1244	Success	0.1212	Success
Linear complexity	0.2321	Success	0.2119	Success



a. Rössler system b. Duffing Oscillator

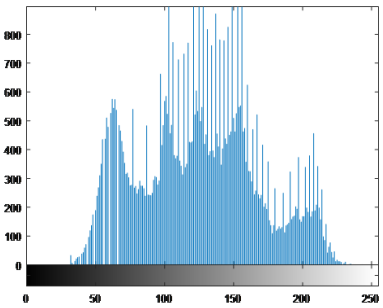
Figure 5.1. Histogram of generated random numbers in scaled range $(0, 1)$

pixels of the original image are presented. For fully description of the algorithm, the reader should refer to [51]. For our analysis, we use Lena image with size 256x256. Firstly, we use Rössler system as PRNG in encryption algorithm[51]. The ciphered and original images of the Lena by their corresponding histogram analysis is shown in Fig.5.2.b for Rössler system as PRNG. According to histogram analysis, ciphered image (Fig. 5.2.d) displays behavior close to uniform distribution so that according to histogram analysis, Rössler system can be a good candidate for image encryption as PRNG. In Fig 5.2.f, histogram analysis results for Duffing oscillator is used as CPRNG for image encryption is shown. It also displays similar performance to the Rössler system. In addition, ciphered images, deciphered images are given in Fig. 5.3.a and Fig. 5.3.c with their corresponding histogram analysis. According to Fig. 5.3, we can successfully reverse the encryption operation on the ciphered images.

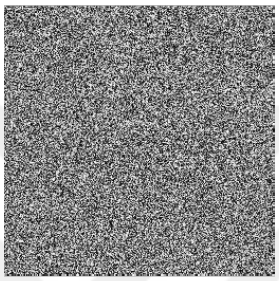
Entropy E gives information about the degree to which system is random. If the entropy of the system is high, the encryption is more secure. For our Lena image, maximum entropy value can be 8. For cipher image done by using Rössler system, the entropy E is 7.9928 which very close to optimum value so our encryption is secure against entropy attacks. On the other hand, Duffing oscillator has an entropy values $E = 7.9916$ which is also very close to optimum value but slightly lower than performance of Rössler system.

In the correlation analysis, the correlation between adjacent pixels is examined. For security purposes cipher image should have very low correlation coefficients. We test cipher

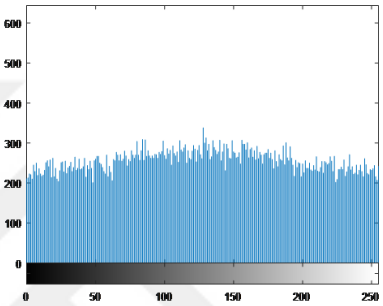
a. Original Lena image



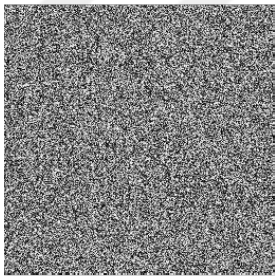
b. Histogram of Lena image



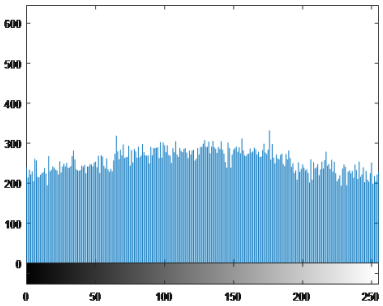
c. Ciphered image for Lena using the Rössler system system



d. Histogram of cipher image for Lena using Rössler system



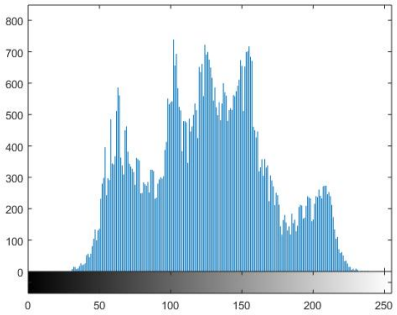
e. Ciphered image for Lena using the Duffing oscillator



f. Histogram of cipher image for Lena using the Duffing oscillator

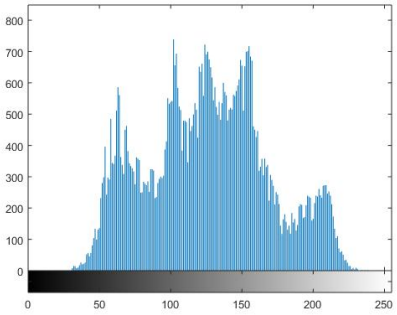
Figure 5.2. Cipher and original images of Lena and their histograms

a. Deciphered image for Lena using the Rössler system



b. Histogram of deciphered image for Lena using Rössler system

c. Deciphered image for Lena using the Duffing oscillator



d. Histogram of deciphered image for Lena using Duffing oscillator

Figure 5.3. Deciphered images and their histogram analysis

images generated by Rössler and Duffing oscillator systems in the following cases: vertical, horizontal and diagonal adjacent pixels.

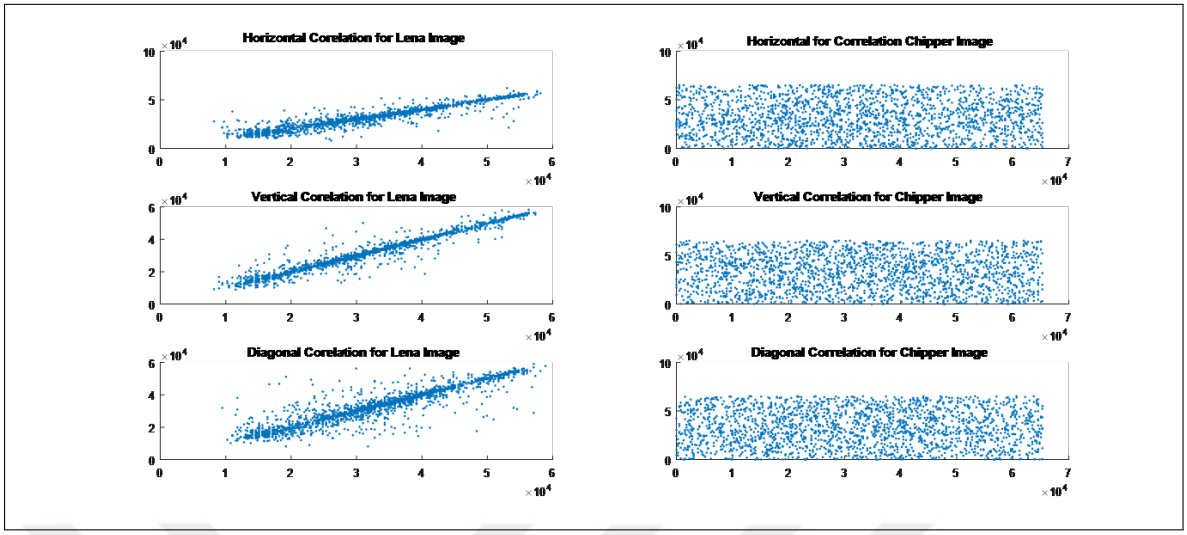


Figure 5.4. Correlation analysis of Lena Figure and its cipher image by Rössler system

Table 5.2. Correlation Coefficients for Adjacent pixels

Correlation Coefficients	**Lena Image**	**Cipher Image by Rössler system**	**Cipher Image by Duffing Oscillator**
Horizontal	0.9607	0.0125	0.0211
Vertical	0.9276	0.0753	0.0761
Diagonal	0.9057	0.0561	0.0541

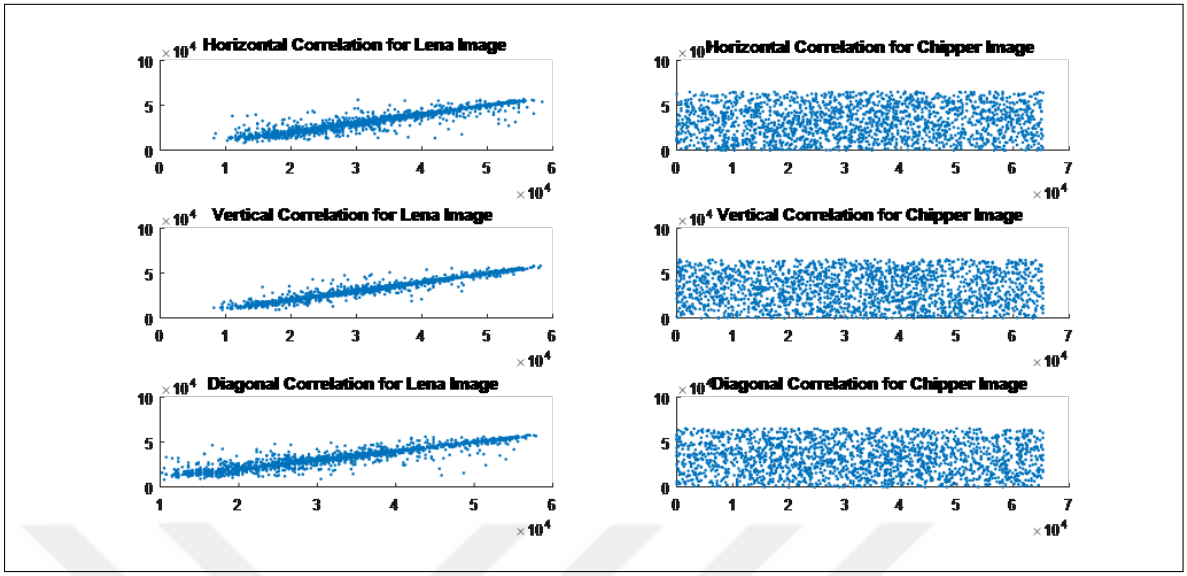


Figure 5.5. Correlation analysis of Lena Figure and its cipher image by Duffing system

6. CONCLUSION

In this dissertation, we worked on three different problems. Firstly, we analyzed the MYMH system (Eq. 3.1) which is a very important model to understand current evolution of universe. Secondly, we made use of neural networks to make prediction about nonlinear systems. We use neural networks not just to predict nonlinear time series but also to estimate Lyapunov exponents and modeling dynamical systems. Finally, we constructed a random number generator from two different dynamical systems which can be very useful for image encryption and Monte-Carlo applications in physics.

We have studied the MYMH system and have tried to investigate the possible chaotic behavior of the MYMH system based on following the effect of parameter g on the stability of the system. We observed that for small values of energy E system display chaotic behavior where there exist two isolated islands around a fixed point $(0, 0, 0, 0)$. When we increase the value of the parameter g, the stabilizing effect of the harmonic oscillator term of the Hamiltonian becomes obvious. After application of the method of averaging, we have observed that changing values of parameter g can result in Hopf bifurcation. We find this result important since we can construct a positive Lyapunov function whose derivative gives exactly zero. Thus, we have found the existence of a possible approximately conserved quantity and asymptotic structural stability in the system.

In chapter 4, we compare and test the performance of three neural network architecture for prediction of chaotic time series generated from dynamical systems. We observe very simple network structures give very efficient estimation of nonlinear phenomena compared to other well-known techniques such as least squares estimates. According to our observations, their success originated from their ability in nonlinear function estimation. They are also very trustful tool for prediction of chaotic parameters such as Lyapunov exponents as we reported.

In chapter 5, we have demonstrated that our chaotic PRNGs are able to pass statistical tests to be accepted as good real random number generators. In addition to this, we demonstrate their efficiency in image encryption algorithm which is originally based on iterative logistic

map. One possible handicapped situation of our PRNGs is that if the initial conditions and parameter values are known, they can be hacked. To deal with this issue for each instantiation initial conditions can be stated by TRG based on hardware. However, to make use of their chaotic nature as PRNG, certain initial conditions and parameter values should be settled.

REFERENCES

1. Devaney RL, Eckmann JP. An introduction to chaotic dynamical systems. *Physics Today*. 1987;40:72.

2. Gaiko V. Chaos transition in the Lorenz system. *Visnyk Odesk Nats Univers Mat i Mekh*. 2013 10;18:51–58.

3. Vadasz P. Analytical prediction of the transition to chaos in Lorenz equations. *Applied Mathematics Letters*. 2010;23(5):503 – 507.

4. Rugh SE. Chaos in the Einstein equations — characterization and importance? In: *Deterministic chaos in general relativity*. Boston, MA: Springer US; 1994; 359-422. .

5. Poincaré JH, Goroff DL. *New methods of celestial mechanics*. History and Philosophy of Modern Physics. New York, NY: AIP; 1993.

6. Koopman B. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*. 1931;17:315.

7. v Neumann J. Zur operatorenmethode in der klassischen mechanik. *Annals of Mathematics*. 1932;33(3):587–642.

8. v Neumann J. Einige satze uber messbare abbildungen. *Annals of Mathematics*. 1932;33(3):574–586.

9. Haken H. *Advanced synergetics: instability hierarchies of self-organizing systems and devices*. Berlin, Heidelberg: Springer; 1983.

10. Wolf A, Swift JB, Swinney HL, Vastano JA. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*. 1985;16(3):285 – 317.

11. Lyapunov AM, Walker JA. The general problem of the stability of motion. *Journal of Applied Mechanics*. 1994;61:226.

12. Osedelec VI. A multiplicative ergodic theorem: characteristic Lyapunov exponents of dynamical systems. *Transaction on Moscow Mathematical Society*. 1968;19:197–231.

13. Ruelle D. Ergodic theory of differentiable dynamical systems. *Publications Mathématiques de l'Institut des Hautes Études Scientifiques*. 1979 12;50(1):27–58.

14. Wiesel W. Continuous time algorithm for Lyapunov exponents. I. *Physical Review E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*. 1993 06;47:3686–3691.

15. Wiesel W. Continuous time algorithm for Lyapunov exponents. II. *Physical Review E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*. 1993 06;47:3692–3697.

16. Rosenstein MT, Collins JJ, Luca CJD. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*. 1993;65(1):117 – 134.

17. Kantz H. A robust method to estimate the maximal Lyapunov exponent of a time series. *Physics Letters A*. 1994;185(1):77–87.

18. Gencay R, Dechert WD. An algorithm for the n Lyapunov exponents of an n-dimensional unknown dynamical system. *Physica D: Nonlinear Phenomena*. 1992;59(1):142–157.

19. Maus A, Sprott JC. Evaluating Lyapunov exponent spectra with neural networks. *Chaos, Solitons and Fractals*. 2013;51:13–21.

20. Gencay R. Nonlinear prediction of noisy time series with feedforward networks. *Physics Letters A*. 1994;187(5):397–403.

21. Maguire LP, Roche B, McGinnity TM, McDaid LJ. Predicting a chaotic time series using a fuzzy neural network. *Information Sciences*. 1998;112(1):125–136.

22. Gencay R, Liu T. Nonlinear modelling and prediction with feedforward and recurrent networks. *Physica D: Nonlinear Phenomena*. 1997;108(1):119 – 134.

23. Maus A, Sprott JC. Neural network method for determining embedding dimension of a time series. *Communications in Nonlinear Science and Numerical Simulation*. 2011;16(8):3294 – 3302.

24. Krylov NM, Bogoliubov NN. New method of nonlinear mechanics in their application to the investigation of the operation of electronic generators. *I United Scientific and Technical Press: Moscow*. 1934;9:109–115.

25. Krylov NM, Bogoliubov NN. *Introduction to non-Linear mechanics. (AM-11)*. Annals of Mathematics Studies. Princeton: Princeton University Press; 2016.

26. Guckenheimer J, Holmes P. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. Applied Mathematical Sciences. New York: Springer; 2013.

27. Goldstein H, Poole CP, Safko JL. *Classical mechanics*. San Francisco: Addison Wesley; 2002.

28. Hale JK. *Ordinary differential equations*. Dover Books on Mathematics Series. Mineola, New York: Dover Publications; 2009.

29. Meyer K, Hall G. *Introduction to Hamiltonian dynamical systems and the n-body problem*. Applied Mathematical Sciences. New York: Springer; 2013.

30. Rand RH. *Topics in nonlinear dynamics with computer algebra*. Newark, NJ, USA: Gordon and Breach Science Publishers, Inc.; 1994.

31. G Matinyan S, K Savvidi G, G Ter-Arutyunyan-Savvidi N. Stochasticity of classical Yang-Mills mechanics and its elimination by using the Higgs mechanism. *Jetp Letters - JETP LETT-ENGL TR*. 1981 01;34.

32. Vigo-Aguiar M, Sansaturio M, Ferrándiz J. Integrability of Hamiltonians with polynomial potentials. *Journal of Computational and Applied Mathematics*. 2003;158(1):213–224.

33. Yoshida H. Non-integrability of the truncated Toda lattice Hamiltonian at any order. *Communications in Mathematical Physics*. 1988;116(4):529–538.

34. Bountis T, Segur H, Vivaldi F. Integrable Hamiltonian systems and the Painlevé property. *Physical Review A*. 1982;25(3):1257–1264.

35. La Salle J, Lefschetz S, Alverson RC. Stability by Liapunov's direct method with applications. *Physics Today*. 1962;15:59.

36. Casdagli M. Nonlinear prediction of chaotic time series. *Physica D: Nonlinear Phenomena*. 1989;35(3):335 – 356.

37. de Oliveira KA, Vannucci A, da Silva EC. Using artificial neural networks to forecast chaotic time series. *Physica A: Statistical Mechanics and its Applications*. 2000;284(1):393 – 404.

38. Hegger R, Kantz H, Schreiber T. Practical implementation of nonlinear time series methods: TISEAN package. *Chaos: An Interdisciplinary Journal of Nonlinear Science*. 1999;9(2):413–435.

39. Pislaru C, Shebani A. Identification of nonlinear systems using radial basis function neural network. *International Journal of Computer, Information, Systems and Control Engineering*. 2014;8(9):1528–1533.

40. Boussaada Z, Curea O, Remaci A, Camblong H, Mrabet Bellaaj N. A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation. *Energies*. 2018;11(3).

41. Andrecut M. Logistic map as a random number generator. *International Journal of Modern Physics B*. 1998;12(09):921–930.

42. Kanso A, Ghebleh M. A novel image encryption algorithm based on a 3D chaotic map. *Communications in Nonlinear Science and Numerical Simulation*. 2012;17(7):2943–2959.

43. Vajargah BF, Asghari R. A pseudo random number generator based on chaotic henon map (CHCG). *IJMEC*. 2015;5(15):2026–37.

44. Lynnyk V, Sakamoto N, Čelikovskỳ S. Pseudo random number generator based on the generalized Lorenz chaotic system. *IFAC-Papers OnLine*. 2015;48(18):257–261.

45. Šenkeřík R, Pluháček M, Viktorin A, Komínková Oplatková Z. On the simulation of complex chaotic dynamics for chaos based optimization. *Proceedings-30th European Conference on Modelling and Simulation, ECMS 2016*; 2016. p. 258–264.

46. Canals V, Morro A, Rosselló JL. Random number generation based on the Rossler attractor. *IEICE Proceeding Series*. 2014;1:272–275.

47. Merah L, Ali-Pacha A, Said NH, Mamat M. A pseudo random number generator based on the chaotic system of Chua's circuit, and its real time FPGA implementation. *Applied Mathematical Sciences*. 2013;7(55):2719–2734.

48. Shannon CE. A mathematical theory of communication. *Bell System Technical Journal*. 1948;27(3):379–423.

49. Rukhin A, Soto J, Nechvatal J, Barker E, Leigh S, Levenson M, et al.. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Gaithersburg, MD, US: *NIST Special Publication*; 2001.

50. Wan Tsang W, Marsaglia G. Some difficult-to-pass tests of randomness. *Journal of Statistical Software*. 2002;7(2):1–9.

51. Kr Banthia A, Tiwari N. Image encryption using pseudo random number generators. *International Journal of Computer Applications*. 2013;67(4):1–8.

APPENDIX A: Recurrent Neural Network Matlab Code

```
function [ffnet,Y1,performance,trset]=elmannet(y,m,eta)
y=y(:);
y=y';
[nyr,nyc]=size(y);
trset=100;
trset=floor(0.8*nyc*trset/100);
yL=lagmatrix(y',1:m)';
lr = maxlinlr(yL(:,m+1:end),'bias');
PRL=min(yL')';
PRU=max(yL')';
ffnet=newelm([PRL(1:m) PRU(1:m)],[2*m 1],{'tansig','purelin'},'trainlm');
 ffnet.trainparam.epochs=1000;
 ffnet.trainparam.goal=1e-9;
 ffnet=train(ffnet,yL(1:m,m+1:trset),y(1,...
    m+1:trset));
A=yL(1:m,trset+1:nyc);
A=A+eta*sqrt(var(y))*randn(size(A));
Y1 = sim(ffnet,A);
performance=perform(ffnet,y(1,trset+1:nyc),Y1);
t=trset+1:nyc;
plot(t,Y1,'.');
hold on
plot(t,y(1,trset+1:nyc));
xlim([min(t) max(t)]);
xlabel('Time');
ylabel('X(t)');
legend('estimated','original');
end
```

Figure A.1. Recurrent Neural Network Matlab Code

APPENDIX B: Multilayer Feedforward Neural Network Matlab Code

```
function [ffnet,Y1,performance,trset]=feedmultilayer(y,m,eta)
y=y(:);
y=y';
[nyr,nyc]=size(y);
trset=100;
trset=floor(0.8*nyc*trset/100);
yL=lagmatrix(y',1:m)';
lr = maxlinlr(yL(:,m+1:end),'bias');
PRL=min(yL')';
PRU=max(yL')';
ffnet=newff([PRL(1:m) PRU(1:m)],[m 2*m m 1],{'tansig','tansig','tansig'...
      ,'purelin'},'trainlm');
 ffnet.trainparam.epochs=2000;
 ffnet.trainparam.goal=1e-9;
 ffnet=train(ffnet,yL(1:m,m+1:trset),y(1,...
    m+1:trset));
A=yL(1:m,trset+1:nyc);
A=A+eta*sqrt(var(y))*randn(size(A));

Y1 = sim(ffnet,A);
performance=sqrt(perform(ffnet,y(1,trset+1:nyc),Y1));
t=trset+1:nyc;
plot(t,Y1,'.');
hold on
plot(t,y(1,trset+1:nyc));
xlim([min(t) max(t)])
xlabel('Time');
ylabel('X(t)');
legend('estimated','original');
end
```

Figure B.1. Multilayer Feedforward Neural Network Matlab Code